

Grado Universitario en Ingeniería Telemática
Curso académico 2017-2018

Trabajo Fin de Grado

Diseño y prototipado de un sistema de comunicaciones para un monoplaza de Formula Student

Rubén Moreno Carrillo

Tutor

Borja Genovés Guzmán

21 de Septiembre de 2018, Leganés, Madrid



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**

RESUMEN

Los sistemas de telemetría en los vehículos de competición son el nexo entre los mundos de las comunicaciones y la competición. Todo aficionado de las carreras, sobre todo de coches, conoce la importancia de los sistemas de telemetría en estas disciplinas. En el caso de la competición, aunque no es el único campo en el que se utilizan, se tratan de sistemas capaces de enviar en tiempo real datos que reflejan el comportamiento del vehículo en tiempo real a un equipo colocado fuera del mismo, para que el equipo pueda monitorizar el estado del vehículo.

El objetivo de la presente memoria es reflejar las pautas que se han llevado a cabo durante el desarrollo de un sistema de telemetría específico para el monoplaza que representa a la Universidad Carlos III de Madrid en la competición universitaria *Formula Student*.

Para determinar con qué herramientas contar para el desarrollo del proyecto, se ha realizado un estudio comparativo con diferentes soluciones. Además, se han llevado a cabo pruebas para comprobar que las decisiones tomadas eran acertadas.

Por último, se realizará una valoración del resultado final, analizando tanto sus aspectos positivos como los negativos, para finalizar enumerando posibles acciones para mejorar el sistema de telemetría en el futuro.

Palabras clave: telemetría; formula, competición; transceptor; radio; antenas; Arduino; Shield; nRF24L01.

ABSTRACT

Telemetry systems in competition vehicles are the connection between communications and competition worlds. Every race fan, specially car race supporters, knows the importance of the telemetry systems in these disciplines. Competition is not the only area where these systems are used, but in this case, teams use them to send to the team's box real time data from the car.

The objective of this document is to show all the steps that have been followed during the development of a specific telemetry system for the Carlos III University team race car, participant in Formula Student competition.

To determine which devices we have to consider to develop this project, a comparative research about different possible solutions has been realized. Furthermore, tests have been done to check if the made decisions were the correct ones.

Finally, in the conclusions paragraph we will valorate the final result, analizing the positive and the negative points, to then finish describing possible future actions to improve this telemetry system.

Key words: telemetry; formula, competition; transceiver; radio; antennas; Arduino; Shield; nRF24L01.

*Si has llegado hasta aquí
no te puedes rendir ahora.*

AGRADECIMIENTOS

Mi hermano Mis padres Luís Inclán Sánchez Mi tutor

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.	x
ÍNDICE DE TABLAS	xii
ÍNDICE DE ABREVIATURAS	xiii
1. INTRODUCCIÓN.	1
1.1. Definición de objetivos	1
1.2. Planteamiento del problema.	2
1.2.1. El equipo Formula UC3M.	2
1.3. Entorno socio-económico	4
1.3.1. Impacto socio-económico	4
1.3.2. La competición <i>Formula Student</i>	5
1.4. Marco regulador	6
1.4.1. Organización de las bandas de radiofrecuencia	7
1.4.2. Bandas ISM.	8
2. ESTADO DEL ARTE.	11
2.1. Introducción a la telemetría	11
2.2. Aplicaciones actuales	12
2.2.1. La telemetría en competición	13
2.3. Sistemas de telemetría similares	15
3. DISEÑO DEL SISTEMA Y JUSTIFICACIÓN DE LA SOLUCIÓN	17
3.1. Lista de sensores	19
3.2. Elección de plataforma	23
3.3. Elección de tecnología.	26
3.4. Elección de dispositivos	33
3.5. Soportes para el sistema	36
3.5.1. Diseño.	36
3.5.2. Ubicación	36
4. DESARROLLO DEL SISTEMA	44
4.1. Programación del emisor	45

4.2. Programación del receptor.	45
4.3. Interfaz de visualización.	46
5. PRUEBAS REALIZADAS.	49
5.1. Pruebas de los módulos de radio	49
5.1.1. Prueba de diferentes antenas	50
5.1.2. Prueba de alcance	54
5.2. Resultados en la competición	60
6. PLANIFICACIÓN DEL TRABAJO.	62
7. PROBLEMAS ENCONTRADOS Y POSIBLES SOLUCIONES	66
7.1. Unidad de control de motor (ECU).	66
7.2. Sensores de efecto Hall	66
7.3. Situación de la antena en el monoplaza	66
7.4. Pruebas en competición	68
8. PRESUPUESTO.	69
9. CONCLUSIONES	72
9.1. Conclusiones generales	72
9.2. Futuras líneas de trabajo.	72
BIBLIOGRAFÍA	
ANEXO A: PRESUPUESTOS SOLUCIONES COMERCIALES.	
ANEXO B: CÓDIGO ARDUINO.	
ANEXO C: CÓDIGO ARDUINO PRUEBAS	
ANEXO D: ENGLISH SUMMARY	

ÍNDICE DE FIGURAS

1.1	Equipo <i>Formula UC3M</i> en la competición de <i>Formula Student Spain</i> 2017	3
1.2	Tipos de pruebas y su puntuación	6
1.3	División de las tres regiones establecidas por la ITU en la superficie terrestre (Región 1: Amarillo; Región 2: Azul; Región 3: Morado)	8
2.1	Esquema del sistema de telemetría en un Fórmula Uno	14
2.2	Hoja de telemetría subida por el piloto Lewis Hamilton a Twitter	15
3.1	Circuito de <i>Formula Student Czech</i> en el autódromo de Most (República Checa)	17
3.2	Circuito de <i>Formula Student Spain</i> en el circuito de Montmeló (Barcelona, España)	18
3.3	Sensores de los amortiguadores	19
3.4	Sensores de temperatura de los frenos	20
3.5	Sensores de temperatura de rodamientos y circuito de refrigeración	21
3.6	Potenciómetro para medir la posición de los pedales	22
3.7	Efecto Hall	22
3.8	Tubo de Pitot	23
3.9	Variedad de placas Arduino	37
3.10	Módulos XBee	38
3.11	Paquete transmitido por el nRF24L01 [11]	38
3.12	Campo de control de paquetes (<i>Packet Control Field</i>) [11]	38
3.13	Modulación FSK	38
3.14	Filtro Gaussiano	39
3.15	Placa Arduino Uno	39
3.16	Placa Arduino Mega	40
3.17	Placa Arduino Nano	40
3.18	Módulo nRF24L01	41
3.19	Módulo tarjeta SD	41
3.20	Arduino Mega Sensor Shield	42

3.21	Primer escenario de la prueba de alcance	43
4.1	Módulos del sistema colocados en el coche	44
4.2	Esquema del montaje en el coche	45
4.3	Captura de pantalla de la herramienta Team Viewer	46
4.4	Diagrama de flujo del código del módulo emisor	47
4.5	Diagrama de flujo del código del módulo receptor	48
5.1	Antenas A (11 cm) y B (21 cm)	49
5.2	Analizador de red HP Agilent 8753E	50
5.3	Generador de señales HP Agilent N5181A	51
5.4	Analizador de espectros HP 8594E	52
5.5	Prueba realizada en el laboratorio	52
5.6	Posición de las antenas para la primera prueba realizada en el laboratorio .	53
5.7	Primer escenario de la prueba de alcance	55
5.8	Gráfica de número de paquetes recibidos a diferentes distancias (escenario 1)	57
5.9	Gráfica del FLR en el escenario 1	57
5.10	Segundo escenario de la prueba de alcance	58
5.11	Gráfica de número de paquetes recibidos a diferentes distancias (escenario 2)	59
5.12	Gráfica del FLR en el escenario 2	60
5.13	Equipo <i>Formula UC3M</i> en la competición de <i>Formula Student Spain</i> 2018	61
6.1	Diagrama de Gantt del proyecto	65
7.1	Posición final de la antena en el monoplaza	67
9.1	C3-R18 prototype car	

ÍNDICE DE TABLAS

1.1	Puntuación en las pruebas de <i>Formula Student</i>	6
1.2	Bandas de frecuencia dentro del espectro radioeléctrico	8
1.3	Bandas ISM	10
5.1	Resultados prueba medición del parámetro S_{11}	53
5.2	Resultados prueba potencia recibida	54
5.3	Resultados prueba con y sin regulador	56
5.4	Número de paquetes recibidos a diferentes distancias (escenario 1)	56
5.5	FLR en el escenario 1	56
5.6	Número de paquetes recibidos a diferentes distancias (escenario 2)	58
5.7	FLR en el escenario 2	59
8.1	Costes de personal	69
8.2	Coste de material	70
8.3	Costes totales	71

ÍNDICE DE ABREVIATURAS

BER: Tasa de Error de Bit(*Bit Error Rate*).

CEPT: Conferencia Europea de Administraciones de Correos y Telecomunicaciones (*European Conference of Postal and Telecommunications Administrations*).

CNAF: Cuadro Nacional de Atribución de Frecuencias.

CRC: Control de Redundancia Cíclica(*Cyclic Redundancy Check*).

ECU: Unidad de Control de Motor (*Engine Control Unit*).

FIA: Federación Internacional del Automóvil (*Federation Internationale de l'Automobile*).

FLR: Tasa de pérdida de tramas(*Frame Loss Ratio*).

Formula SAE: Competición entre estudiantes organizada por la Sociedad de Ingenieros del Automóvil(*Society of Automotive Engineers*).

FSK: Modulación por Desplazamiento de Frecuencia (*Frequency Shift Keying*).

GFSK: Modulación por Desplazamiento de Frecuencia Gaussiana (*Gaussian Frequency Shift Keying*).

GPIO: Entrada/Salida de Propósito General (*General Purpose Input/Output*).

GSM: Sistema Global para las Comunicaciones Móviles(*Global System for Mobile communications*).

IDE: Entorno de Desarrollo Integrado (*Integrated Development Environment*).

ISM: (*Industrial, Scientific and Medical*). Bandas de radiofrecuencia reservadas para uso no comercial en áreas industrial, científica y médica.

ITU: Unión Internacional de Telecomunicaciones (*International Telecommunication Union*).

LTE: (*Long Term Evolution*).

OFDM: Multiplexación por División de Frecuencias Ortogonales (*Orthogonal Frequency-Division Multiplexing*).

SPI: (*Serial Peripheral Interface*).

TSC: Teoría de la Señal y Comunicaciones.

UMTS: Sistema Universal de Telecomunicaciones Móviles (*Universal Mobile Telecommunications System*).

USRP: Dispositivos de radio definidos por software (*Universal Software Radio Peripheral*).

1. INTRODUCCIÓN

La motivación del desarrollo de este proyecto viene originada por la unión de dos grandes aficiones: las telecomunicaciones y la competición dentro del mundo de la automoción. Teniendo la oportunidad de poder unir la pasión por el mundo del motor con la elaboración del proyecto probablemente más importante de la etapa de un estudiante, sólo cabe aprovecharla y fusionar ambas disciplinas para disfrutar con el desarrollo de este proyecto. Esta posibilidad surgió al ingresar en el equipo *Formula UC3M*, participante en la competición universitaria *Formula Student*.

En el presente documento se presentará el sistema de telemetría desarrollado para el coche construido por el equipo que representa a la Universidad Carlos III de Madrid en esta competición internacional. Este sistema cubre la necesidad de monitorizar todos los sensores instalados en el coche con el objetivo de seguir escalando posiciones dentro de la clasificación general de equipos y conseguir mejorar la fiabilidad y la seguridad de su monoplace de competición.

1.1. Definición de objetivos

El principal objetivo es el desarrollo e implementación de un sistema de telemetría en tiempo real para ayudar al equipo a obtener un mejor resultado en las competiciones de 2018, no sólo en el rendimiento dinámico de su monoplace, sino también en las pruebas estáticas de la competición que se explicarán más adelante en el apartado 1.3.2.

El sistema de telemetría deberá ser capaz de recopilar y mostrar en tiempo real, en un dispositivo independiente alejado del vehículo, tanto los datos recogidos por una serie de sensores colocados a lo largo de todo el coche como los datos guardados en la Unidad de Control de Motor, o *Engine Control Unit* (ECU). Estos datos mostrarán la temperatura, velocidad, revoluciones o posición de distintas piezas para controlar que esos valores están dentro de unos márgenes de seguridad, ayudando a mejorar tanto la vida útil de las piezas como la seguridad del propio piloto.

Una de las principales motivaciones del desarrollo de este sistema es la prevención de roturas de piezas del vehículo. Gracias a la monitorización de la temperatura de ciertos elementos críticos del coche se podrá conocer instantáneamente su situación, permitiendo en caso de sobrecalentamiento avisar al piloto para que pare y apague lo antes posible el motor, evitando así posibles roturas o averías, mejorando así la seguridad del piloto también.

Aunque el principal objetivo sea mejorar la fiabilidad y la seguridad en el vehículo, para poder realizar simulacros de competición antes de llevar el coche a competir, el desarrollo de este sistema también pretende utilizar los datos recopilados para su posterior

análisis. El fin de este estudio será mejorar tanto el rendimiento y la elección de los pilotos, comparando sus habilidades y su evolución en las distintas pruebas, como la configuración y ajuste inicial del coche, utilizando, por ejemplo, los datos recogidos del comportamiento de los amortiguadores.

También se busca que el sistema tenga un rango de alcance que abarque sin problemas la distancia entre el punto más alejado del circuito donde se compite y la posición donde va a poder estar un miembro del equipo para recibir los datos en su ordenador.

En última instancia, lo que se persigue con el desarrollo e implementación de este sistema, es mejorar el coche de años anteriores y así continuar evolucionando y creciendo dentro de la competición, demostrando a los jueces que año tras año se suman nuevos avances en el diseño del prototipo y el equipo consigue ser cada vez más competitivo.

1.2. Planteamiento del problema

En años anteriores el coche no disponía de sistema de telemetría. Por ello, tanto en las pruebas previas como en la competición se tenía el miedo de que cualquier elemento pudiera romperse. No hay que olvidar que no deja de ser un coche diseñado y fabricado por alumnos de la Universidad con la ayuda de algunos profesores. Por este motivo, los coches de *Formula Student* suelen tener poca fiabilidad. Además, no se contaba con datos que reflejaran el comportamiento dinámico del coche, haciendo que tareas como la de calibrar la suspensión dependiera totalmente de las sensaciones del piloto.

Para solucionar este problema se planteó desarrollar un sistema de telemetría en el equipo para poder mitigar en la medida de lo posible este problema, colocando sensores para controlar la evolución de ciertas piezas del coche mientras está en pista y para obtener datos que puedan ayudar a desarrollar el buen comportamiento del monoplaza. Uno de los principales objetivos al desarrollar este sistema es evitar el sobrecalentamiento de piezas como los rodamientos, frenos o componentes del motor.

1.2.1. El equipo Formula UC3M

Esta asociación nació en el año 2011, gracias a las ganas y entusiasmo de un grupo de alumnos de nuestra Universidad por llevar a cabo una empresa tan complicada, y ambiciosa a priori, como es fabricar un monoplaza de competición.

Desde ese mismo año se trabajó muy duro, realizando un gran esfuerzo y trabajo de investigación sobre la competición, su duro reglamento, el trabajo de los equipos que participaban en ella y la búsqueda de apoyos económicos.

Con escasos recursos y apenas ayuda por parte de la Universidad se consiguió el ansiado objetivo: desarrollar el primer coche de *Formula Student*, El C3-R16, diseñado enteramente por los miembros del equipo y fabricado prácticamente en su totalidad dentro de las instalaciones del campus de Leganés en el año 2016, con la ayuda en algunos

procesos de fabricación por parte de sus patrocinadores.

El equipo siempre ha competido en la categoría de motores de combustión, aunque como objetivo a medio plazo se ha establecido competir en la categoría eléctrica.

Dentro de este proyecto convergen alumnos de distintas disciplinas de grado, máster y doctorado, tanto de ramas tecnológicas como de humanidades, que contribuyen con sus conocimientos obtenidos a lo largo de sus estudios y adquiridos durante su participación dentro del equipo en años anteriores. Gracias a esta diversidad, el equipo se nutre de recursos humanos para sus distintas áreas de trabajo o departamentos. Pero no sólo está formado por alumnos de la Universidad. También cuenta con la ayuda y apoyo de algunos profesores, cuya contribución es vital para asesorar, enseñar y colaborar con los estudiantes, además de hacer de nexo entre la asociación y las instituciones de la Universidad.

Los departamentos trabajan interconectados para facilitar la comunicación entre ellos y contribuir en todas las tareas de diseño, simulación y fabricación del monoplaza. El equipo está compuesto por 39 estudiantes de hasta 7 titulaciones diferentes, haciendo de él un equipo multidisciplinar y competente para enfrentar su objetivo y todos los retos y adversidades que se van presentando durante el curso.



Fig. 1.1. Equipo *Formula UC3M* en la competición de *Formula Student Spain* 2017

Hasta ahora sólo se han desarrollado dos coches, uno en 2016 y otro en 2017. En ambos casos sólo se ha competido a nivel nacional, en la competición *Formula Student Spain*, disputada en el circuito internacional de Montmeló, situado en Barcelona. En la Fig. 1.1 se pueden ver a los miembros del equipo que acudieron a la competición de 2017.

Este año se pretende dar el salto internacional para competir en la competición *For-*

mula Student Czech, disputada en República Checa, además de repetir un año más en la edición nacional con el prototipo que se está desarrollando este año, el C3-R18.

En mi caso, formo parte del equipo desde principios del curso académico 2016/2017. Como integrante del departamento de electrónica, mi trabajo durante estos dos años, además de ayudar en las diferentes tareas que se han desarrollado por parte de mi departamento y en las labores generales del equipo como puede ser la fabricación del coche, ha sido diseñar e implementar un sistema de telemetría capaz de transmitir en tiempo real los datos de los sensores alojados en el coche a un ordenador que esté situado en una zona alejada de donde esté circulando el vehículo.

Una vez este sistema esté instalado, ayudará a conocer con mayor detalle el comportamiento real del coche antes de la competición, permitiendo probar distintas configuraciones de sus distintos elementos con la intención de mejorarlo.

1.3. Entorno socio-económico

En el caso de este proyecto concreto el entorno socio-económico queda acotado dentro del mundo de la competición automovilística. Pese a ello, la telemetría como herramienta para monitorizar datos de sensores en tiempo real está cobrando cada vez más presencia debido a la gran evolución de la tecnología conocida como Internet de las cosas.^o *Internet of Things* (IoT) [1].

Para el llamado IoT, los sistemas de monitorización utilizarán Wi-Fi o la nueva tecnología del 5G [2], entre otras, aprovechando su baja latencia y consumo eléctrico para transmitir los datos.

Este tipo de sistemas permiten ahorrar tiempo y costes, evitando tener que acudir al emplazamiento donde están colocados los sensores para recoger sus datos. Así, por ejemplo, podrían monitorizarse en tiempo real las condiciones climáticas a las que se encuentran plantaciones agrícolas, o las medidas de consumo de agua en un edificio sin que un técnico tenga que acercarse hasta la finca para realizar la misma medida.

Volviendo al área de la competición, y más concretamente a nuestro caso, *Formula Student*, los equipos siempre tienen como principal objetivo año tras año la reducción de peso y la mejora de la fiabilidad de sus coches, siempre con la menor inversión económica posible. Por ello, es muy importante conseguir desarrollar un sistema de telemetría que ayude a mejorar la fiabilidad del coche, que sea lo más ligero posible y que pueda ser reutilizable en futuros años, consiguiendo así un uso sostenible de los recursos.

1.3.1. Impacto socio-económico

Si analizamos el impacto socio-económico del proyecto, cabe decir que con una inversión económica bastante reducida en materiales y herramientas, se puede conseguir un sistema de telemetría funcional que ayude en las labores de mantenimiento y mejora del

monoplaza de un equipo competidor en *Formula Student*, además de mejorar la seguridad tanto del conductor como de todas las personas que puedan estar dentro del radio de acción del vehículo.

En el caso de un equipo de competición profesional, el hecho de contar con un sistema de telemetría a medida marca la diferencia. Con él pueden comprobar si el desarrollo de nuevos elementos en su coche han funcionado o no, midiendo su eficacia de manera efectiva y objetiva, permitiéndoles ahorrar tiempo para poder invertirlo en nuevas mejoras.

Aunque los objetivos no sean exactamente los mismos, el desarrollo de este nuevo sistema de telemetría permitirá al equipo *Formula UC3M* la posibilidad de escalar puestos dentro de la clasificación, debido a que puede obtener un mayor número de puntos en las pruebas de la competición relacionadas con la innovación y el diseño del coche. Este hecho permitirá dar más visibilidad tanto nacional como internacional al equipo, aumentando las posibilidades de conseguir nuevos patrocinadores, y por tanto, un mayor número de ingresos económicos.

1.3.2. La competición *Formula Student*

Formula Student, también conocida como *Formula SAE (Society of Automotive Engineers)*, es una competición que tiene su origen en la Universidad de Houston en el año 1979. Creada como una competición entre universidades ha ido creciendo hasta convertirse en la denominada "*Fórmula Uno universitaria*".

Su objetivo es que los estudiantes utilicen todos los conocimientos y habilidades que han adquirido durante sus estudios aplicándolos en un proyecto que simula uno real, como es la fabricación de un coche de competición. La competición evalúa las fases más técnicas de un proyecto real de estas características como es la investigación, el diseño, la fabricación y las pruebas dinámicas, pero también evalúa la fase de administración y finanzas, teniendo que diseñar incluso un plan de marketing.

Se celebran ediciones de este tipo de competiciones a lo largo de todo el mundo, aunque algunas de las más importantes son celebradas en Estados Unidos, Australia, Japón, India, Alemania, Inglaterra o España.

La competición simula la siguiente situación: una empresa de competición busca un equipo de diseño para desarrollar un coche de tipo Fórmula. El equipo debe presentar tanto su diseño como un estudio económico para garantizar que tras la fabricación de una serie de unidades el proyecto continúa siendo rentable. Por ello el equipo debe esforzarse en todos los aspectos que en el futuro podrán encontrarse en su vida laboral.

La competición se divide en 3 categorías (motor eléctrico, motor de combustión y coches autónomos) y evalúa el trabajo de los equipos participantes en pruebas estáticas y dinámicas. Pero antes de poder participar en dichas pruebas, todos los coches deben superar una serie de inspecciones técnicas por parte de los jueces para garantizar la seguridad del evento. A parte de revisiones oculares, todos los monoplazas deben superar una

prueba de frenado, una de vuelco y una prueba de ruido antes de poder competir.

Las pruebas puntuables de la competición se indican en la Tabla 1.1, y el balance entre los dos tipos de pruebas existentes en la Fig. 1.2 . Como es de suponer, todas las pruebas dinámicas consisten en terminar el recorrido en el menor tiempo posible.

Pruebas estáticas		325 Puntos
Plan de negocio o <i>Bussiness Plan</i>	Estudio económico del proyecto	75
Análisis de costes	Presentación del coste total del proyecto	100
Diseño	Evaluación de las decisiones de ingeniería tomadas	150
Pruebas dinámicas		675 Puntos
<i>Skidpad</i>	Circuito en forma de ocho	75
Aceleración	Recorrido de 75m en línea recta	75
<i>Autocross</i>	Circuito con slaloms	100
<i>Endurance</i>	Recorrer 22km en el circuito de <i>Autocross</i>	325
Eficiencia	Medida del combustible consumido durante <i>Endurance</i>	100
Total		1000 Puntos

Tabla 1.1. Puntuación en las pruebas de *Formula Student*



Fig. 1.2. Tipos de pruebas y su puntuación

1.4. Marco regulador

El sistema de telemetría está constituido por un sistema de comunicaciones inalámbrico que utiliza el espectro radioeléctrico como canal de transmisión. Por ello, debemos estudiar el marco regulatorio referente a las emisiones radioeléctricas.

El espectro radioeléctrico es un bien de dominio público, cuya titularidad y administración corresponden al Estado. En el caso de nuestro país el organismo encargado es el ministerio de Industria, Comercio y Turismo.

En el Cuadro Nacional de Atribución de Frecuencias (CNAF)[3], se recogen los usos reservados a cada una de las bandas de frecuencia en las que se divide el espectro radioeléctrico disponible para radiocomunicaciones, entre 8,3 kHz y 3000 GHz. Esta normativa es una pieza básica para el ordenamiento del espectro y contiene información de carácter técnico sobre la utilización del espectro en las diferentes bandas de frecuencias en España.

Debido a que tanto este año como en el futuro los planes del equipo *Formula UC3M* son poder competir en otras competiciones europeas, se han tenido en cuenta también la regulación europea e internacional del mismo recurso para asegurarnos de que la decisión tomada sea válida y pueda ser utilizada en cualquier competición a la que podamos asistir.

En el ámbito internacional es la *International Telecommunication Union* (ITU) la organización encargada de mantener la cooperación y coordinación entre sus miembros, asignando los servicios asociados a las diferentes frecuencias. Por otro lado, dentro de la Unión Europea es el *European Conference of Postal and Telecommunications Administrations* (CEPT) el organismo que regula el uso del espectro radioeléctrico.

Las reglas de *Formula Student* no especifican ningún apartado en el que se regulen los sistemas de transmisión de datos, y por tanto no hace referencia a ningún protocolo a utilizar. Esto da toda la libertad a los equipos para que utilicen el que mejor se adapte a sus necesidades.

En el siguiente apartado se presentará las diferentes bandas de frecuencias existentes y sus principales usos.

1.4.1. Organización de las bandas de radiofrecuencia

Con el objetivo de garantizar una utilización de manera óptima del espectro radioeléctrico destinado a las comunicaciones este se divide en bloques llamados bandas de frecuencias. A cada una de estas bandas se les atribuye una serie de servicios, exclusivos o compartidos.

El espectro radioeléctrico comprende desde los 3 kHz de frecuencia, con una longitud de onda de 100 kilómetros, hasta los 30 GHz, con una longitud de onda de 1 milímetro. En la Tabla 1.2 se muestra la división del espectro radioeléctrico en las diferentes bandas de radio, con sus respectivas frecuencias y longitudes de onda.

Cuanta más baja sea la frecuencia, mayor es su longitud de onda y su alcance; mientras que si aumentamos la frecuencia, su longitud de onda también disminuye y su alcance es menor.

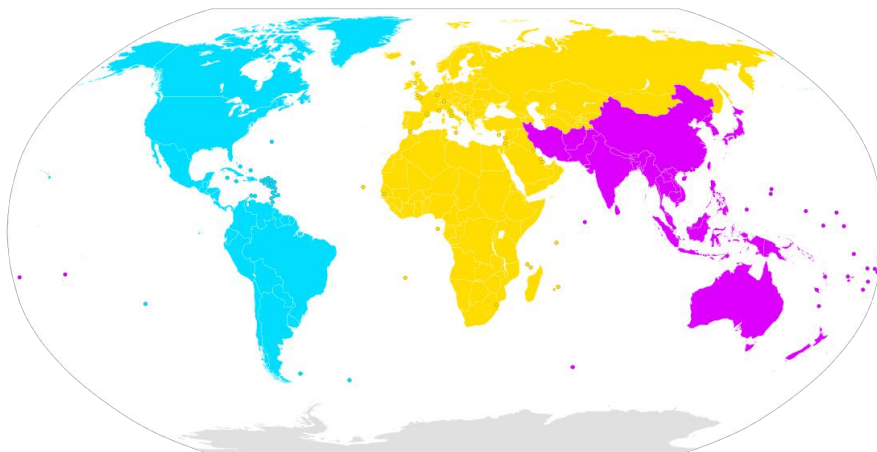
Hay que tener en cuenta que muchas de estas asignaciones dependen de la tecnología a utilizar. Por ejemplo, los radares más avanzados requieren frecuencias más altas, mientras que los dispositivos para la navegación submarina operan en frecuencias bajas. Así también, existen franjas reservadas para el uso militar o para los servicios de emergencia. Algunas de estas frecuencias están delimitadas por la normativa internacional.

Bandas de frecuencia	Frecuencias	Longitudes de onda
Banda VLF (<i>Very Low Frequencies</i>)	3 - 30 kHz	100 - 10 km
Banda LF (<i>Low Frequencies</i>)	30 - 300 kHz	10 - 1 km
Banda MF (<i>Medium Frequencies</i>)	300 - 3000 kHz	1000 - 100 m
Banda HF (<i>High Frequencies</i>)	3 - 30 MHz	100 - 10 m
Banda VHF (<i>Very High Frequencies</i>)	30 - 300 MHz	10 - 1 m
Banda UHF (<i>Ultra High Frequencies</i>)	300 - 3000 MHz	100 - 10 cm
Banda SHF (<i>Super High Frequencies</i>)	3 - 30 GHz	10 - 1 cm
Banda EHF (<i>Extremely High Frequencies</i>)	30 - 300 GHz	10 - 1 mm

Tabla 1.2. Bandas de frecuencia dentro del espectro radioeléctrico

1.4.2. Bandas ISM

Las bandas ISM (*Industrial, Scientific and Medical*) son bandas de radiofrecuencia reservadas para uso no comercial en áreas industrial, científica y médica. Estas bandas fueron definidas por la ITU-R (el sector de Radiocomunicaciones de la ITU). Se pueden utilizar sin necesidad de licencia siempre que se cumplan unos determinados límites de potencia. Algunas de ellas son las utilizadas por tecnologías tan populares como Wi-Fi o Bluetooth.



[Imagen]. Recuperado de

https://es.wikipedia.org/wiki/Regi%C3%B3n_de_la_Uni%C3%B3n_Internacional_de_Telecomunicaciones

Fig. 1.3. División de las tres regiones establecidas por la ITU en la superficie terrestre (Región 1: Amarillo; Región 2: Azul; Región 3: Morado)

A lo largo de todo el mundo existen tres regiones diferentes establecidas por la ITU con el propósito de administrar el espectro electromagnético de manera global. Cada una de las regiones, las cuales se diferencian en la Fig. 1.3, posee su propio conjunto de asignación de frecuencias.

En cuanto a la potencia de emisión permitida en España, para las emisiones que utilizan estas bandas de frecuencia, su límite está establecido en 500 mW para evitar posibles daños a cualquier persona, como recoge el Boletín Oficial del Estado (BOE) del día 09/05/2013, actualizado el día 27/10/2017 [4].

En la Tabla 1.3 se muestran las bandas ISM definidas por la ITU-R. Estas bandas difieren en algunos casos entre las tres regiones. Están diferenciadas por una doble línea dentro de la tabla entre ellas. La primera pertenece a la región VHF; la segunda a la región UHF; y la tercera pertenece a la región de microondas, la cual va desde 1 GHz hasta 300 GHz.

Una de las desventajas que tiene utilizar frecuencias dentro de las bandas ISM es que muchísimos dispositivos también las utilizan, aumentando la probabilidad de sufrir interferencias. Por ello los dispositivos se suelen diseñar con gran selectividad en frecuencia y muy buena calidad a nivel de enlace.

Rango de frecuencias		Tipo	Disponibilidad	Servicio con licencia
6,765 MHz	6.795 MHz	A	Sujeto a regulación local	Servicios móviles y fijos
13,553 MHz	13.567 MHz	B	Mundial	Servicios móviles y fijos, excepto servicios móviles de radiocomunicaciones aeronáuticas
26.957 MHz	27.283 MHz	B	Mundial	Servicios móviles y fijos, excepto servicios móviles de radiocomunicaciones aeronáuticas, radio BC*
40.66 MHz	40.7 MHz	B	Mundial	Servicios móviles y fijos y servicio de exploración de la Tierra por satélite
433.05 MHz	434.79 MHz	A	Sólo en región 1, sujeto a regulación local	Radio amateur y servicio de radiolocalización
902 MHz	928 MHz	B	Sólo en región 2, con algunas excepciones	Servicios móviles y fijos, excepto servicios móviles de radiocomunicaciones aeronáuticas y servicio de radiolocalización. En región 2, servicio amateur adicional
2.4 GHz	2.5 GHz	B	Mundial	Servicios fijos y móviles , amateur y servicio satelital amateur
5.725 GHz	5.875 GHz	B	Mundial	Servicios fijos, móviles, amateur, de radiolocalización y servicio satelital amateur
24 GHz	24.25 GHz	B	Mundial	Servicios amateur, satelital amateur, de radiolocalización y servicio de exploración de la Tierra por satélite (activo)
61 GHz	61.5 GHz	A	Sujeto a regulación local	Servicios fijos, móviles, inter satelitales y de radiolocalización
122 GHz	123 GHz	A	Sujeto a regulación local	Servicios fijos, móviles, de exploración de la Tierra por satélite (pasivo), inter satelitales de investigación del espacio (pasivo) y servicios amateur
244 GHz	246 GHz	A	Sujeto a regulación local	Servicios de radiolocalización, radio-astronomía, amateur y servicio satelital amateur

*Radio BC = Radio de banda ciudadana

Tipo A: El uso de estas bandas de frecuencia ISM estarán sujetas a la autorización especial por parte de la administración que le concierna. Las administraciones deberían seguir las últimas recomendaciones ITU-R.

Tipo B: Los servicios de radiocomunicación que operen en estas bandas deberán aceptar posibles perjuicios ocasionados por interferencias con otros servicios ISM.

Tabla 1.3. Bandas ISM

2. ESTADO DEL ARTE

El problema queda planteado de forma clara. El planteamiento del problema incluye productos, desarrollos o trabajos similares, o referencias bibliográficas. Los referentes con los que se compara son muy apropiados, están bien descritos, y se realiza una comparación entre las diferentes soluciones presentadas.

2.1. Introducción a la telemetría

La telemetría es el sistema que permite la medición remota de magnitudes físicas y químicas, y el envío de los datos recogidos hacia una central de control para su análisis y provecho.

Normalmente se utiliza la comunicación inalámbrica pero también se puede realizar a través de soluciones cableadas como redes de ordenadores o fibra óptica, entre otras tecnologías.

El sistema de telemetría típicamente está formado por un transductor, que actúa como dispositivo de entrada; un medio de transmisión, que bien pueden ser ondas de radio o un sistema cableado; dispositivos de procesamiento de la señal y un dispositivo de grabación y/o visualización de la señal. El transductor convierte la magnitud medida, como la temperatura, presión, nivel o velocidad, en una señal eléctrica que es transmitida a una determinada distancia hasta la central de control para proceder a su visualización o análisis.

El proceso completo desde que el sistema recoge los datos del sensor o los sensores hasta que la persona encargada visualiza esos datos está compuesto por tres fases: la adquisición de datos, la transmisión y el análisis.

- En la primera fase actúan varios agentes: el o los sensores que recogerán la información que queremos medir y el transductor, encargado de recabar esa información de los sensores y convertirlos en una señal eléctrica. Además, en la mayoría de los casos se suele añadir un sistema de almacenamiento que guarde todos los datos a modo de copia de seguridad para asegurar que ningún dato se pierde a consecuencia de un fallo o rotura en cualquiera de las otras dos fases.
- Dependiendo del tipo de transductor que utilicemos tendrá la capacidad de enviar los datos por el canal de comunicación elegido o necesitará la ayuda de algún otro dispositivo para la fase de transmisión.
En este último caso será necesario que el transductor que ha recibido los datos de los sensores y el transmisor estén comunicados entre sí para que este último haga de puente entre el transductor y el receptor.

- Y en la última fase, la de visualización y análisis, entran en escena los dispositivos que recogen e interpretan la señal recibida y el software que nos muestra la información de manera que nosotros la podamos entender.

2.2. Aplicaciones actuales

La telemetría es utilizada en una gran variedad de aplicaciones de diversas áreas como pueden ser el automovilismo, aviación, astrología, agricultura, industria, medicina o biología entre otras.

Dentro del área industrial las principales aplicaciones se dan en la industria aeronáutica y aeroespacial, para monitorizar los parámetros de aeronaves no tripuladas; en plantas químicas, para controlar procesos que pueden llegar a ser peligrosos; en redes de suministro, para llevar un control de la cantidad de electricidad, agua y gas producida y consumida; y en la industria de la automoción, especialmente en el campo de la competición.

Además de en todos los campos mencionados anteriormente, el uso de sistemas de telemetría se está abriendo paso en otras áreas como la agricultura, la medicina, la domótica, la biología o el área del IoT.

A continuación se explicarán algunas de estas aplicaciones en diversos campos.

Telemetría vehicular

Permite conocer el rendimiento del motor, la presión de los neumáticos o el aceite, la eficiencia aerodinámica o las temperaturas de piezas importantes y de elementos vitales como la del aceite en el motor o el agua en el sistema de refrigeración. Todas estas medidas resultan claves para conocer la progresión y el estado del vehículo durante la conducción.

Telemetría en medicina

Existen multitud de aparatos de medición de parámetros médicos que, a través de una conexión por medio de línea telefónica o por conexión mediante el puerto serial a un ordenador con conexión a internet, son capaces de mandar los resultados al sistema central para la grabación de esos datos en el historial clínico del paciente. En muchos casos estos dispositivos son utilizados en casos en los que el paciente tiene dificultades para acudir a un centro médico o en los que es necesario hacer una monitorización durante 24 horas del paciente, por ejemplo realizar un registro electrocardiográfico para registrar anomalías durante el día. Con este sistema el paciente puede estar en su casa, aumentando su comodidad y evitando su hospitalización durante ese día.

Algunos de los aparatos que pueden llevar incorporado este sistema son el electrocardiógrafo, tensiómetro, oxímetro, espirómetro, analizador de orina o analizador de sangre.

Telemetría en la agricultura

Aunque parezca que los mundos de la agricultura y de las telecomunicaciones estén muy alejados, existen sistemas de telemetría para controlar los niveles de humedad, temperatura o presión barométrica de las plantaciones. Existen incluso sistemas de alerta que avisan a los agricultores mediante el servicio de mensajes cortos o *Short Message Service* (SMS) si alguno de los niveles medidos por los sensores se encuentra fuera de los límites dentro de los que idealmente la plantación debería estar.

Telemetría espacial

Las misiones espaciales no serían posible sin los sistemas de telemetría. Gracias a ellos se pueden obtener mediciones y verificaciones de los satélites, naves tripuladas o sondas entre otros. Estas medidas son de suma importancia para la seguridad de todas estas misiones, ya que sin ellos no se tendría constancia de la situación de los elementos enviados una vez efectuado su despegue.

Telemetría en el Internet de las cosas

El Internet de las cosas consiste en la interconexión de multitud de elementos cotidianos como nuestros coches, electrodomésticos o incluso las luces del alumbrado público de nuestras ciudades a través de Internet. En algunos de estos casos, como el alumbrado público o el termostato de nuestras casas, el objetivo de establecer esta conexión a Internet es poder controlar su estado remotamente, analizar los datos que aportan o visualizarlos.

2.2.1. La telemetría en competición

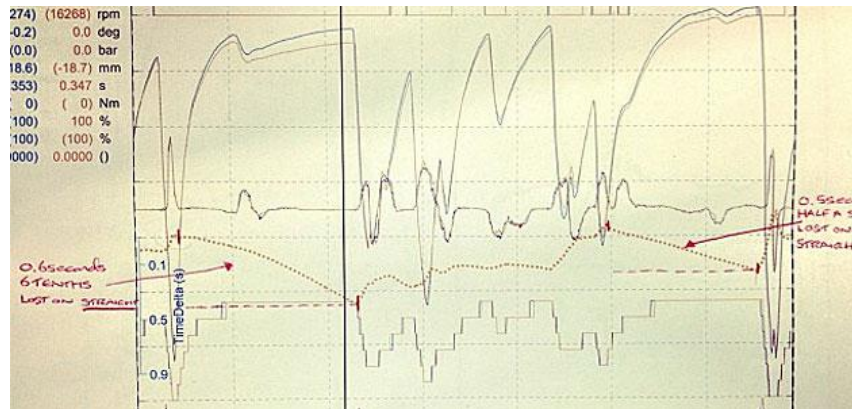
El uso más conocido de la telemetría se asocia con la competición automovilística. Dentro de ella, se ha convertido en uno de los sistemas más importantes de los que se dispone, ya que los datos se utilizan para determinar la configuración del coche.

El principal proveedor de sistemas de telemetría para las grandes competiciones de velocidad, como Fórmula Uno, MotoGP o el mundial de rallies (World Rally Championship, WRC), es el fabricante Magneti Marelli, que ofrece soluciones de conectividad, almacenamiento de datos y sistemas electrónicos. Pero Magneti Marelli no sólo provee a equipos de competición, también ofrece soluciones para particulares tanto de los ya mencionados como de otros sistemas de coches o motocicletas, como los de suspensión, escape o transmisión.

En la Fórmula Uno, la competición automovilística por excelencia, todos los equipos cuentan con un sistema de telemetría que establece una comunicación entre el coche y el ordenador del ingeniero de pista. Utilizan microondas a una frecuencia de entre 1.45 y 1.65 GHz o la frecuencia permitida por las autoridades locales, dentro de la banda UHF

datos recopilados gracias a la telemetría, ya que es la principal herramienta que utilizan el desarrollo de su coche.

Por otro lado, también es conocido el uso de la telemetría en el otro gran campeonato mundial del motor: MotoGP. Pero en realidad estos sistemas no son telemetría como tal. En todas las competiciones de motociclismo están prohibidas las transmisiones de datos de manera inalámbrica en tiempo real.



[Imagen]. Recuperado de <https://albrodpulf1.wordpress.com/2014/03/07/analisis-telemetria/>

Fig. 2.2. Hoja de telemetría subida por el piloto Lewis Hamilton a Twitter

Por este motivo, en todas estas competiciones los llamados sistemas de telemetría realmente sólo realizan la primera fase de la telemetría, que consiste en la recogida y almacenamiento de todos los datos de los sensores.

Estos sistemas son llamados *dataloggers*. Almacenan toda la información recogida por la ECU para su posterior análisis por parte de los ingenieros de cada equipo una vez que la moto está ya en el box.

En realidad son sistemas similares a los utilizados en la Fórmula Uno, salvo por la diferencia de que mientras que en la competición automovilística los datos son enviados al equipo en tiempo real, en el caso de MotoGP y sus categorías inferiores el equipo debe esperar a que la moto vuelva al box para descargar toda la información recogida en esa sesión. Los datos y software para visualizarlos también son similares.

En ambos casos, estos sistemas también son utilizados por los jueces de estas competiciones para estudiar las causas de cualquier accidente y determinar si fue un fallo humano, mecánico, intencionado o involuntario para imponer, en el caso que fuera necesario, las correspondientes sanciones.

2.3. Sistemas de telemetría similares

Con la comparativa de proyectos similares se busca conocer las decisiones que otros autores han tomado ante problemas similares y compararlos con nuestras soluciones, pu-

diendo encontrar así un apoyo a la hora de tomar nuestras decisiones.

Para tener una comparativa dentro de la misma competición, se conocen las tecnologías utilizadas por algunos equipos. Por ejemplo, las Universidades de Zurich, Waterloo, Barcelona, Valencia o Bolonia utilizan tecnología Wi-Fi; el equipo de la Universidad Politécnica de Madrid ha optado por ZigBee; los de las Universidades de Western Australia y Bratislava utilizan tecnologías GSM; y otros equipos como el de la Universidad de Delft utilizan protocolos propios de radio no estandarizados.

Algunos de los trabajos con objetivos similares a los nuestros que hemos encontrado son los siguientes:

- **TELEMETRÍA DE VEHÍCULOS FSAE CON PIC32 Y TRANSCEPTOR DE RADIO CC110L [5]**

En este trabajo el autor utiliza la tecnología radio para la transmisión de los datos, pero se centra más en la lectura de los datos de la ECU por el Bus CAN, mientras que nosotros transmitiremos datos de sensores colocados en distintos puntos del coche.

- **SISTEMA DE TELEMETRIA PARA UN VEHÍCULO DE FORMULA S.A.E. [6]**

En este caso los módulos de radio elegidos para la transmisión son los transceptores XBee. También transmite los datos leídos desde la ECU por el Bus CAN, pero además, el autor diseña una interfaz de visualización con la herramienta LabVIEW.

- **DESARROLLO DE UN DISPOSITIVO DE TELEMETRIA Y GEOLOCALIZACION BASADO EN LA PLATAFORMA ARDUINO Y SHUIELD 3G+GPS [7]**

Los objetivos de este trabajo son enviar los datos a través de Internet utilizando cobertura 3G y conseguir desarrollar un sistema que ofrezca la posición del coche utilizando un módulo con GPS.

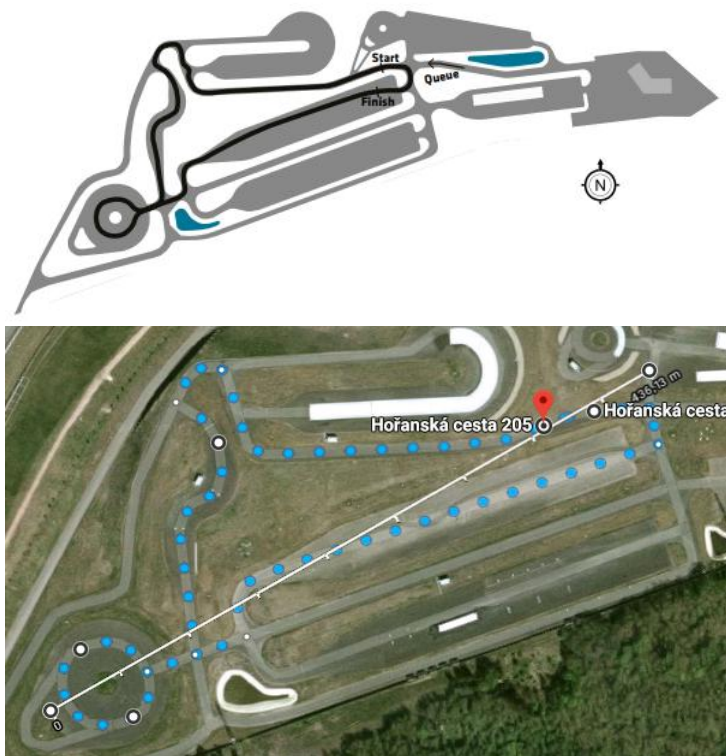
- **DISEÑO DE UN SISTEMA DE TELEMETRIA PARA UN MONOPLAZA DE FORMULA STUDENT [8]**

De nuevo los módulos elegidos para el desarrollo de este proyecto son los módulos XBee para transmitir los datos recopilados por la ECU. En este caso también se ha diseñado una interfaz de visualización con la herramienta LabVIEW.

3. DISEÑO DEL SISTEMA Y JUSTIFICACIÓN DE LA SOLUCIÓN

Cuando por primera vez se planteó diseñar el sistema, aún existían muchas incógnitas por resolver. No se conocía el presupuesto final para el desarrollo del sistema de telemetría pero se sabía que iba a ser muy ajustado. Tampoco se sabía el número total de sensores que iban a ser colocados en el coche ni de cuáles de ellos era importante conocer sus medidas en tiempo real. De este modo no se conocía ni rango de alcance necesario del sistema, ni ancho de banda ni presupuesto del que partir para diseñar nuestro prototipo.

Aunque no se sabía si este año se iba a participar en más de una competición, si se sabía que al menos el equipo iba a competir como en años anteriores en *Formula Student Spain* (FSS), y como existía la posibilidad de competir fuera de España, se tomó la decisión de implementar un sistema que fuera cual fuera la tecnología de comunicación que utilizara pudiera ser utilizada en el extranjero, más concretamente en Alemania y República Checa, que eran las competiciones a las que el equipo iba a intentar inscribirse.

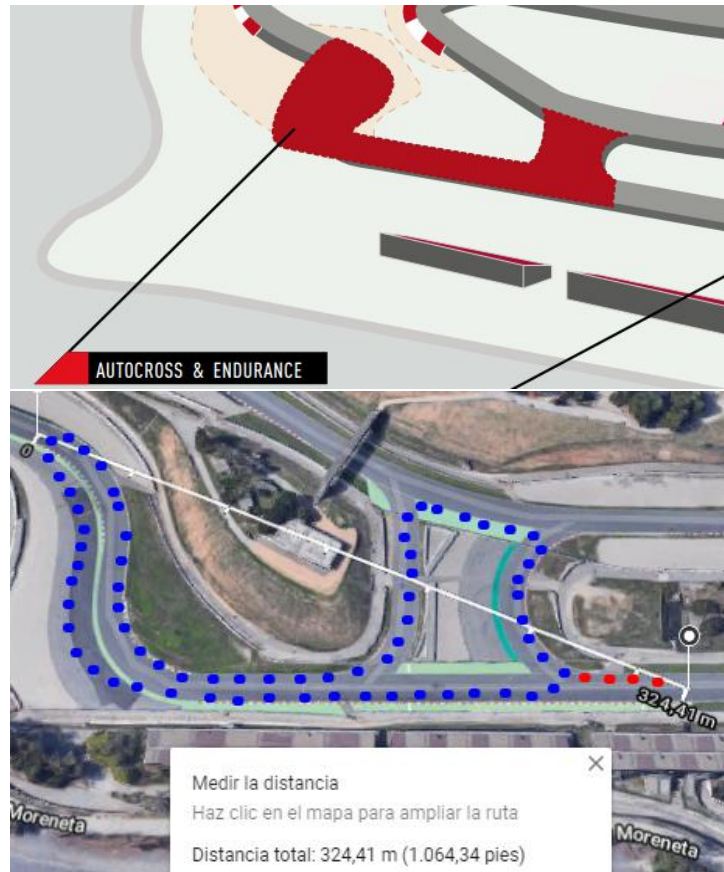


Puntos más alejados del circuito: aproximadamente 440 metros.

Imágenes obtenidas del *Handbook* de FSCzech y de Google Maps. Recuperado de <https://www.fsczech.cz/docs/Handbook2018.pdf>.

Fig. 3.1. Circuito de *Formula Student Czech* en el autódromo de Most (República Checa)

Para acotar algo más el alcance necesario del sistema, se compararon los circuitos de la prueba de *Autocross* y *Endurance*, que son los más largos en los que se compite. Como se puede ver en las figuras Fig. ?? y Fig. 3.2 la longitud máxima de punta a punta de los circuitos no supera los 450 metros, teniendo en cuenta que las medidas mostradas representan el peor de los casos.



Puntos más alejados del circuito: aproximadamente 325 metros.

Imágenes obtenidas del *Handbook* de FSS y de Google Maps. Recuperado de <http://formulastudent.es/wp-content/uploads/2017/08/handbook2017-baja.pdf>

Fig. 3.2. Circuito de *Formula Student Spain* en el circuito de Montmeló (Barcelona, España)

Como resultado de este primer estudio preliminar, se ha decidido que un buen objetivo sería intentar desarrollar un sistema con un poco de margen para que su alcance no estuviera demasiado cerca de la distancia mínima que requerimos para así evitar posibles pérdidas o desconexiones en algunas partes del circuito debidas a obstáculos imprevistos, como el desnivel del trazado o los monoplazas del resto de equipos. Así, el primero objetivo es encontrar una tecnología de comunicación cuyo rango efectivo esté alrededor de 1000 metros. Tras revisar las cuentas del equipo, se ha determinado que la compra de todos los elementos necesarios para implementar el sistema de *datalogger* y transmisión de datos no debería superar los 500 €.

La primera fase del diseño es conocer todas las necesidades que tenemos para poder

establecer los objetivos para cubrirlas. Conociendo ya la limitación económica y el alcance que nuestro sistema debe tener, ya sólo falta lo más importante: conocer la cantidad de datos que necesitamos transmitir por unidad de tiempo, y así estimar el ancho de banda necesario para nuestro sistema.

3.1. Lista de sensores

Para poder determinar la cantidad de datos que va a ser transmitida es necesario conocer el número de sensores que llevará colocado el coche. Para ello, se mostrará a continuación la lista con todos los sensores que queremos monitorizar en tiempo real con nuestro sistema de telemetría:

Amortiguadores (x4) *Modelo: KP13S (KA Sensors)*

Estos sensores son de los más importantes de todos los que irán colocados en el coche. Los sensores de la Fig. 3.3 son sensores analógicos que devuelven el valor de la extensión de sus vástagos en milímetros. Acoplados a los amortiguadores delanteros y traseros del coche, recogen su extensión o compresión. Con estos datos podremos conocer cómo se han comportado estos en las curvas. Estos datos son muy importantes porque son los únicos que nos sirven para conocer la estabilidad del coche tanto en rectas como en curvas. La estabilidad es un aspecto muy importante del coche, ya que determina su manejabilidad y se puede ganar mucho tiempo por vuelta.



Fig. 3.3. Sensores de los amortiguadores

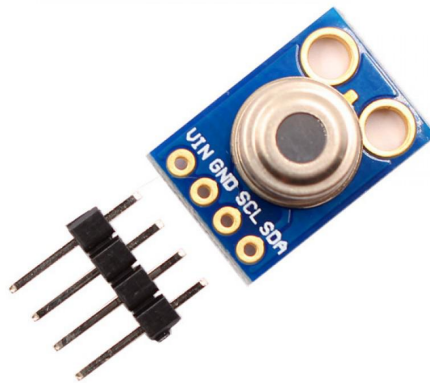
Conociendo cómo se ha balanceado el coche en cada momento es más fácil conocer si la configuración de los amortiguadores es la correcta. Si la dureza y el rebote de los

amortiguadores no están bien configurados el coche no tendrá un buen comportamiento en los giros, haciendo que el piloto tenga la sensación de que el coche tiende a salirse con más facilidad de la pista.

La estabilidad también es importante en las rectas, ya que el comportamiento de los amortiguadores afecta a la posición del chasis en las aceleraciones, afectando a la tracción que pueden llegar a conseguir los neumáticos contra el asfalto.

Temperatura de frenos (x4) Modelo: MLX90614

Se tratan de sensores de infrarrojos digitales (Fig. 3.4) que ofrecen una señal PWM como salida. La temperatura de los frenos es importante para conocer si la presión de las pinzas de freno es equivalente en ambos lados del coche. En el caso de que un lado frenara más que otro uno de los discos de freno se calentaría más que el otro. Por tanto, estos sensores ayudarán a regular el frenado del coche.



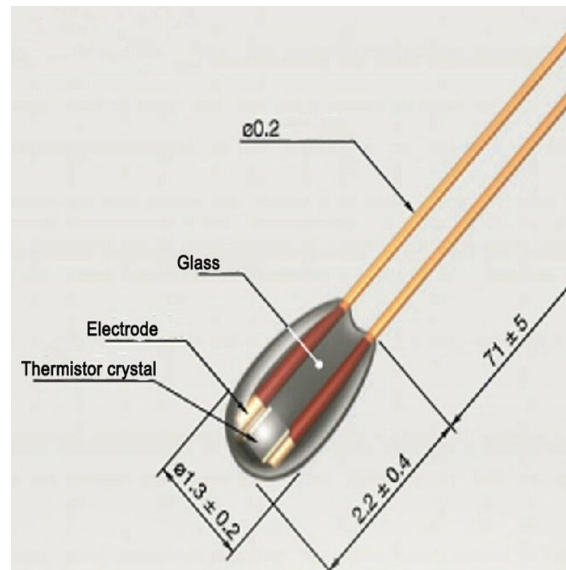
[Imagen]. Recuperado de https://www.makerfabs.com/index.php?route=product/productproduct_id=81

Fig. 3.4. Sensores de temperatura de los frenos

Además es importante conocer la temperatura que han alcanzado los frenos, ya que tanto una temperatura baja como una alta en ellos afecta a su eficacia de frenado. También nos ayudará a conocer la refrigeración que reciben con el aire, permitiéndonos mejorar la aerodinámica en los próximos años para redireccionar más o menos aire hacia los frenos.

Temperatura de rodamientos de manguetas traseras (x2) Termistor común de 100 kiloohmios

Las manguetas son las piezas que unen los bujes de las ruedas (pieza anclada directamente a los discos de frenos y a las llantas) con los conjuntos de suspensión y de dirección en el caso de las ruedas delanteras. Los sensores elegidos para medir la temperatura de los rodamientos que hacen que gire el buje contra la mangueta son unos termopares comunes, cuya estructura se puede observar en la Fig. 3.5.



[Imagen]. Recuperado de <http://aprendiendo.laconeccion.com/producto/termistor-ntc-100k>

Fig. 3.5. Sensores de temperatura de rodamientos y circuito de refrigeración

Es necesario medir la temperatura de estos elementos para evitar sobrecalentamientos y posibles roturas de la propia mangueta o del buje, haciendo imposible la continuidad del equipo en la competición, ya que tanto el buje como la mangueta son piezas diseñadas por miembros del equipo y no se cuenta con piezas de repuesto.

Temperatura del circuito de refrigeración (x3) *Termistor común de 100 kiloohmios*

Con estos sensores se pretende conocer el funcionamiento del sistema de refrigeración, compuesto por el circuito de agua y dos radiadores. Con los datos que recojan se podrá conocer la temperatura del agua antes de pasar por los radiadores y tras atravesarlos. Así podremos saber si los radiadores están haciendo bien su función o no, además de controlar que la temperatura del motor se mantenga siempre dentro de un rango seguro para evitar posibles roturas.

Para este cometido se utilizarán los mismos termopares que los colocados para medir la temperatura de los rodamientos de las manguetas traseras.

Posición de pedales (x2) *Potenciómetro común de 100 kiloohmios*

Se colocarán un par de potenciómetros de 100 kiloohmios similares al de la Fig. 3.6, que indicarán cuándo y cuánto se están pisando los pedales de freno y acelerador midiendo la cantidad de giro que sufren los ejes sobre los que pivotan los pedales. Estos datos se utilizarán en futuras pruebas que se realicen para comparar la conducción

Efecto Hall en las ruedas (x4) *Modelo: SP08 Wheel Speed Sensor*



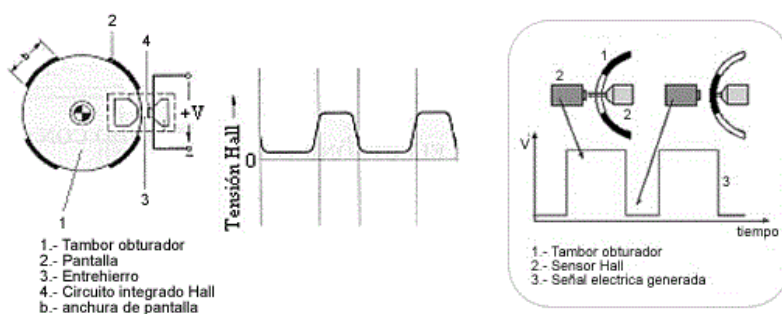
[Imagen]. Recuperado de <https://www.nextiafenix.com/producto/pot-15mm-100k/>

Fig. 3.6. Potenciómetro para medir la posición de los pedales

El efecto Hall es el resultado de la fuerza de Lorentz sufrida por los electrones al estar expuestos a un campo magnético. Los sensores analógicos de efecto Hall están compuestos por un imán fijo, un tambor obturador que anula el campo magnético, y una placa Hall, fabricada por un material conductor y a la cual se le aplica un voltaje en sus extremos.

El tambor gira y se interpone de manera intermitente entre el imán y la placa Hall. Cada vez que el obturador permite la visión directa entre el imán y la placa Hall, la interacción del campo magnético generado por el imán con la corriente eléctrica que transcurre por la placa generará una pequeña fuerza de Lorentz. Dependiendo de la velocidad a la que gire el tambor se generará mayor o menor corriente. Midiendo este valor podemos conocer la velocidad de giro del obturador, que en nuestro caso será la misma a la que gire la rueda.

La Fig. 3.7 muestra visualmente el funcionamiento de estos sensores.



[Imagen]. Recuperado de <http://www.aficionadosalamecanica.net/sensores2-modelos.htm>

Fig. 3.7. Efecto Hall

El dato de la velocidad individual de las ruedas nos permite ajustar el frenado del coche, tanto configurando el reparto de frenada delantero y trasero como ajustando que las ruedas de un mismo eje frenen con la misma intensidad a la vez.

Tubo de Pitot (x1) Modelo: PDA-8-F-6-KL

El tubo de Pitot se utiliza para medir la velocidad del aire a su paso por un punto. Mide la presión del aire cuando pasa a través de él pudiendo calcular su velocidad en ese punto aplicando una fórmula matemática. Midiendo la velocidad del aire a poca distancia de la superficie de los elementos aerodinámicos como los alerones, en las cuales la velocidad del aire debería ser distinta dependiendo de la parte de la pieza en la que coloquemos el sensor de la Fig. 3.8, se estudiará la efectividad de estos elementos desde el punto de vista aerodinámico en los test previos a la competición para cambiar su configuración en los casos que sea posible y mejorar el diseño para años siguientes.



Fig. 3.8. Tubo de Pitot

3.2. Elección de plataforma

Una vez conocidas las limitaciones de presupuesto y distancia que debería cubrir el sistema, queda por determinar qué hardware base y tecnología se utilizarán para su implementación. Para tomar esta decisión se han estudiado y analizado algunas plataformas que pudieran cumplir con nuestros requisitos y nos permitieran unir todos los datos de los sensores en un mismo dispositivo para poder ser enviados desde este.

Soluciones comerciales

La primera opción que se barajó fue buscar una solución comercial que cubriera nuestras necesidades en la fase de envío de la información. De esta manera sólo habría que centrarse en idear la forma de recopilar todos los datos para servírselos al sistema elegido de envío y recepción de esta información, y de gestionar la recogida, tratado y visualización de los datos. Se buscaron una gran cantidad de empresas que ofrecieran sistemas de comunicación inalámbrica, la mayoría del sector industrial y aeronáutico. Los sistemas

buscados eran sistemas de telemetría orientados a otras aplicaciones pero que nosotros pudiéramos adaptar a nuestras necesidades.

Se contactó con 29 empresas para pedirles información sobre sus productos para saber si nos podían ser útiles, o directamente para pedir un presupuesto y ofrecerles un posible patrocinio del equipo para abaratar costes. Entre el listado de empresas a las que se contactó se podían encontrar compañías de todo tipo, desde líderes en el sector de las comunicaciones por radio, como puede ser *Kenwood*, hasta empresas que proveían sistemas de tracking para controlar el comportamiento de aves o empresas que venden sistemas de control para coches de radiocontrol. También se intentó contactar con *Magneti Marelli*, proveedor oficial de los sistemas de telemetría y radio desde 2015 de los campeonatos de Fórmula Uno, MotoGP y WRC. Como curiosidad, tras enviarles varios correos, nunca contestaron.

La gran mayoría de estas empresas satisfacían nuestro objetivo de obtener al menos 1 kilómetro de alcance utilizando frecuencias permitidas en España y Europa, pero el precio de sus productos se alejaba mucho de nuestro presupuesto. En el Anexo A se pueden encontrar un par de presupuestos emitidos por parte de algunas de estas empresas.

Raspberry Pi

Una vez descartada la posibilidad de utilizar soluciones comerciales, se comienza a pensar el diseño de una solución propia para nuestro sistema de telemetría. Para ello, hemos estudiado diferentes soluciones que nos permitan lograr nuestro objetivo.

Raspberry Pi es un mini ordenador de placa reducida de bajo consumo y coste, desarrollado en Reino Unido por la Fundación Raspberry Pi con el objetivo de estimular y hacer alcanzable la enseñanza de la ciencia de la computación y la fabricación digital a lo largo de todo el mundo.

Generalmente este tipo de mini ordenadores ejecutan sistemas operativos basados en Linux y están íntimamente relacionados con el *Open Software*, o software libre, y el *Open Hardware*, o hardware libre.

El software libre son los programas informáticos cuyo código es público y accesible por cualquier persona con el objetivo de que cualquiera pueda utilizarlo y modificarlo para sus propósitos.

El hardware libre es el equivalente en el mundo hardware. La idea es que todas las personas puedan acceder a las especificaciones y diagramas de los dispositivos para poder replicarlos y crear sus propios diseños basados en ellos.

El procesador de una placa Raspberry Pi es más lento que el de un ordenador normal, pero aun así sigue teniendo todas las funcionalidades de un ordenador Linux con un nivel bajo de consumo.

Además de un ordenador, Raspberry Pi incorpora funciones de electrónica como pi-

nes de propósito general o *General Purpose Input/Output* (GPIO), y de comunicación como UART (*Universal Asynchronous Receiver-Transmitter*), y SPI (*Serial Peripheral Interface*), I²C (*Inter-Integrated Circuit*), haciendo posible su integración en proyectos de electrónica como el nuestro, en el cual se pueden incluir mediciones de sensores o comunicación con otras placas.

Arduino

Arduino es una plataforma electrónica o microcontrolador cuyos principios se basan en el software y hardware libres. Es una plataforma flexible y fácil de usar tanto para los creadores como para los desarrolladores que la utilicen.

Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones a través de un código software en un lenguaje determinado. Estos programas permiten interactuar con los circuitos de la placa.

Arduino es un proyecto y no un modelo concreto de placa. Por este motivo podemos encontrar diferentes tipos de placas para distintas aplicaciones, siendo todas ellas una evolución de su placa básica, el Arduino Uno. En la Fig. 3.9 se muestran las placas más populares divididas por ámbitos en los que se utilizan.

Además, las placas Arduino también cuentan con otro tipo de componentes llamados *Shields* o escudos. Se trata de pequeñas placas que se conectan a la placa principal y la complementan añadiéndole una infinidad de funciones, como por ejemplo, conectividad Wi-Fi, Ethernet, GPS o pantallas táctiles LCD.

Arduino es tanto software como hardware, lo que la diferencia de otras placas y microcontroladores. Su principio de hardware libre permite que cualquier otra persona o empresa pueda crear sus propias placas basándose en las desarrolladas por Arduino.

En cuanto al software, Arduino cuenta con un entorno de desarrollo integrado o *Integrated Development Environment* (IDE) propio para casi todas las plataformas (Windows, Linux y Mac), en el cual podemos escribir nuestro código, cargarlo en nuestra placa Arduino y ejecutar o depurar el programa.

USRP (Dispositivos de radio definidos por software)

Los dispositivos reconfigurables de radio definidos por software o *Universal Software Radio Peripheral* (USRP) son transceptores que permiten simplificar el hardware necesario para desarrollar prototipos de sistemas completos de comunicación por radio. Son personalizables mediante software para diseñar dicha comunicación. En este caso se utilizaría *LabVIEW*, una herramienta software de programación gráfica.

Su principal desventaja es el alto coste económico que tiene, razón por la cual hemos descartado utilizar esta opción.

Las principales opciones que destacamos entre todas las analizadas son las de utilizar las plataformas Arduino o Raspberry Pi por su carácter de software libre, versatilidad y facilidad de uso. Dado que ya se cuenta con varias placas Arduino en el equipo desde años anteriores, se ha decidido seguir utilizando Arduino para este proyecto por su compatibilidad y el hecho de que ya se tiene una base de conocimientos utilizando esta plataforma y su IDE para la programación de las placas.

3.3. Elección de tecnología

Una vez conocidas las limitaciones de presupuesto y distancia que debería cubrir el sistema, queda por determinar qué hardware base y tecnología se utilizarán. Aunque el ancho de banda es un aspecto importante a tener en cuenta a la hora de la elección, en nuestro caso es prácticamente despreciable, ya que en total se enviarán únicamente los datos crudos, es decir, el valor numérico de los 20 sensores que irán colocados en el coche. Esta cantidad de datos no superará los 1.5 kB/s, que es una cantidad de datos que las tecnologías que se analizarán a continuación pueden soportar con creces.

Bluetooth

Bluetooth es una tecnología de radio que opera en la banda de frecuencia ISM de los 2.4 GHz. Es utilizada por multitud de servicios de transmisión de datos entre dispositivos, sobre todo de voz. Admite diferentes topologías: punto a punto, difusión o broadcast, y mesh, que permite conectar muchos dispositivos entre sí (many-to-many). En nuestro caso, nos interesaría la topología punto a punto. Además, existe un módulo Bluetooth para la plataforma Arduino, la cual podría ser una opción para desarrollar nuestro sistema.

Una de sus principales virtudes es que no necesita que los dispositivos sean compatible en cuanto a hardware o software. Únicamente hay que establecer un enlace entre los dispositivos, que sin necesidad de configuración previa, sólo tienen que buscarse dentro de su zona de cobertura Bluetooth.

Su alcance únicamente va desde los 10 metros transmitiendo con una potencia de 1 mW hasta los 100 metros con una potencia de transmisión de 100 mW, principal motivo por el que descartamos su utilización.

Wi-Fi

Todos conocemos esta tecnología, acrónimo de *Wireless Fidelity*, ya que la utilizamos diariamente para obtener conexión a Internet en casa, el trabajo, lugares públicos, etc. Se trata del estándar IEEE 802.11, que permite la creación de redes de área local inalámbricas, o WLAN's. Los componentes básicos de una red Wi-Fi son el punto de acceso o AP (*Access Point*), que es la unión entre las redes cableadas y la red Wi-Fi; las antenas conectadas al punto de acceso; y el terminal Wi-Fi, que puede estar conectado a un dispositivo

o bien ya integrado en él.

Como en el caso de Bluetooth, Arduino también ofrece varios módulos para proveer a la placa de conectividad Wi-Fi.

Existen varios estándares dentro del IEEE 802.11, cada uno de ellos orientados a una funcionalidad más específica.

- El estándar 802.11b es el más general de todos, trabaja a 2.4 GHz y transmite a una velocidad de 11 Mbps.
- La primera mejora de velocidad llegó con el 802.11a, que pasó a operar a la frecuencia de 5 GHz, por lo que es incompatible con 802.11b. Utiliza la Modulación por División en Frecuencias Ortogonales o *Orthogonal Frequency Division Multiplexing* (OFDM), que divide la señal en muchas subseñales que son transmitidas simultáneamente al receptor en diferentes frecuencias y permite alcanzar velocidades de transmisión de 54 Mbps.
- El estándar 802.11g es la versión mejorada del 802.11b. También trabaja a 2.4 GHz pero utilizando la misma modulación del 802.11a. Es compatible con 802.11b, permitiendo que puedan trabajar juntos.
- El objetivo del estándar 802.11n es mejorar la velocidad y el alcance de los estándares anteriores. Trabaja en dos bandas de frecuencia, la de los 2.4 y 5 GHz. Es retrocompatible con los estándares anteriormente y su alcance y velocidad se ven mejorados, llegando hasta los 250 metros y hasta los 600 Mbps.
- Ninguno de los estándares anteriores implementa calidad de servicio. Por ello se desarrolló el estándar 802.11e. El objetivo de este estándar es definir los requisitos de los paquetes en cuanto a ancho de banda y retardo de transmisión para permitir mejores transmisiones de audio y vídeo.
- El estándar 802.11i incorporó nuevos mecanismos de autenticación para establecer transferencias de datos más seguras. Este estándar puede cifrar transmisiones en los estándares 802.11a, 802.11b y 802.11g.
- El estándar 802.11p fue ideado para conexiones V2X, que engloba conexiones V2V (vehículo a vehículo) y V2I (vehículo a infraestructura). Trabaja en el espectro de frecuencias de 5.90 GHz y de 6.20 GHz.

De entre todos los estándares Wi-Fi el 802.11p es sin duda el más adaptado a nuestras necesidades. La principal desventaja que presenta frente a otros estándares, además de ser un protocolo aún en desarrollo, es que su seguridad en cuanto a la transferencia de datos es menor, ya que el tiempo de autenticación del protocolo se ha reducido para aumentar el tiempo de transmisión.

Esta desventaja no afecta demasiado a nuestros intereses, ya que en el supuesto de que otro equipo pueda obtener datos en nuestra red local Wi-Fi, algo que intentaremos evitar en la medida de lo posible, los datos que transmitamos es información que nos valen únicamente a nosotros para configurar y controlar el buen estado de nuestro coche.

Además, no somos ni mucho menos un equipo puntero de la competición como para que otros equipos se interesen por nuestros datos. Como mucho podrían sacar información de qué sensores llevamos colocados en el monoplaza, dato que les ofreceríamos sin ningún problema si vinieran a preguntarnos.

WiMax

WiMax es una tecnología muy similar a Wi-Fi pero que utiliza microondas, obteniendo alcances de unos 30 kilómetros y velocidades de hasta 124 Mbps. Sería una opción ideal si no fuera por el gran tamaño de las antenas y el alto precio que tienen los sistemas comerciales.

Además, WiMax es una tecnología como LTE o GSM, las cuáles necesitan cobertura proporcionada por un operador de red. Como no queremos depender de la cobertura que podamos tener tanto en España como en el extranjero, para evitar la posibilidad de que el sistema quede inutilizado por falta de cobertura, hemos decidido desarrollar un sistema que dependa únicamente de sí mismo para funcionar. Por tanto, desestimamos la opción de utilizar WiMax

Red móvil

Otra de las opciones que se han barajado, entre otros motivos porque es la opción elegida por muchos otros equipos para desarrollar su sistema de telemetría, es utilizar la cobertura móvil a través de una tarjeta SIM. Podríamos utilizar redes *Global System for Mobile communications* (GSM), también conocido como cobertura 2G, o las tecnologías *Universal Mobile Telecommunications System* (UMTS) o *Long Term Evolution* (LTE) (3G y 4G respectivamente). En el caso de utilizar GSM, esta tecnología cubriría sin problemas nuestras necesidades. Incluso sería mejor opción que utilizar UMTS o LTE, ya que pese a que estas últimas proporcionan mayor velocidad de transmisión, generalmente ofrecen menor cobertura en zonas no urbanas, como es el caso de los circuitos de velocidad, que suelen estar a las afueras de las ciudades.

Esta opción nos eliminaría el problema de las posibles interferencias y de la distancia entre emisor y receptor. Pero como ya se ha comentado en apartados anteriores preferimos desarrollar un sistema autónomo en cuanto a cobertura para que pueda funcionar en cualquier lugar sin depender de otros agentes.

ZigBee

ZigBee es el nombre de la especificación [9] de un conjunto de protocolos de comuni-

cación por radiofrecuencia digital de bajo consumo. Esta especificación ha sido definida por la *ZigBee Alliance*, una alianza de empresas formada por Texas Instruments, Philips, Freescale y ST entre otros. ZigBee permite el envío de datos, normalmente información de sensores y/o comandos de control, con tasas bajas de transmisión.

Es una tecnología basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal, que define el nivel físico y el control de acceso al medio de redes inalámbricas. Su principal objetivo es desarrollarse en el ámbito de la domótica, debido a su precio económico, su bajo consumo energético que alarga la vida de los dispositivos, su baja tasa de envío de datos y la alta variabilidad en su topología.

Emplea bandas de frecuencias ISM, su alcance depende de la potencia de emisión del dispositivo y su velocidad de transmisión es de hasta 256 kbps.

La forma más práctica de utilizar ZigBee con las placas Arduino es utilizando módulos XBee.

Módulos XBee

Los módulos XBee utilizan el estándar Zigbee para la interconexión y comunicación por radio entre dispositivos. Pertenecen a la familia comercial Digi XBee, desarrollada por *Digi International*, y están diseñados para aplicaciones que requieren un alto tráfico de datos, baja latencia y sincronización en la comunicación.

Existen dos tipos diferentes de hardware básico: XBee serie 1 y XBee serie 2. Se diferencian principalmente en que los módulos XBee serie 2 necesitan ser configurados y en que además de comunicaciones punto a punto y punto-multipunto también soportan redes mesh.

Las redes mesh son redes en las que existen múltiples rutas para alcanzar un destino. Básicamente, permite a dispositivos unirse a la red que a pesar de estar fuera del rango de cobertura de los puntos de acceso están dentro del rango de cobertura de alguna tarjeta de red que, directa o indirectamente, está dentro de la cobertura de un punto de acceso.

Hay que tener en cuenta que la comunicación entre módulos de serie 1 y módulos de serie 2 no es posible.

Para ambas series existen módulos con dos potencias de transmisión diferentes, los XBee y los XBee-PRO, los cuales se muestran en la Fig. 3.10. La versión PRO tiene un consumo eléctrico y precio más elevados pero también incrementa su rango de alcance.

Una de las desventajas que tienen este tipo de transceptores es que, aunque son compatibles con la plataforma Arduino, es necesario un *Shield* o adaptador para conectar el puerto serie de XBee con el de las placas Arduino, incrementando el coste y perjudicando la sencillez de uso.

SigFox

SigFox es una compañía francesa que proporciona un servicio de red de cobertura inalámbrica ideada para el campo del IoT, en el cual los sensores necesitan tener un bajo consumo eléctrico para durar el máximo posible y la tasa de transferencia de datos es muy baja. Por ello, el protocolo que rige Sigfox [10] sólo permite transmitir 12 bytes de datos, pudiendo enviar un máximo de 140 mensajes al día.

Esta compañía propone la creación de una red independiente que requiere incorporar un chip que sea compatible con la misma para poder conectarse. SigFox actúa como un operador que puede hablar directamente con los dispositivos finales, utiliza frecuencias libres ISM resistentes frente a interferencias con cobertura internacional y gestión sencilla y basada en la nube. Desde un único lugar se pueden controlar todos los dispositivos conectados a la misma red.

Con respecto a la cobertura cabe destacar que ofrece cobertura a lo largo de muchos países pero no en toda su extensión, por lo que existen zonas en las que no se podría operar con esta tecnología.

SigFox es una solución que busca la compatibilidad con muchos fabricantes de módulos que adoptan una posición neutra como Avnet, Telit, Texas Instrument, etc. Esto significa que hay más variedad de dispositivos compatibles con esa tecnología en el mercado.

Tras comprobar sus zonas de cobertura, que no cubren los circuitos de las diferentes pruebas europeas de *Formula Student*, y teniendo en cuenta que no permite el envío de datos suficientes para cubrir nuestra necesidad, queda descartado el uso de esta solución orientada a la monitorización de sensores de control para el área del Internet de las cosas.

Otras soluciones que trabajan en bandas ISM

La utilización de tecnologías que operen en bandas de frecuencia dentro de las ISM nos solventaría el problema de tener que utilizar una frecuencia legal tanto en España como en el resto de competiciones en las que podamos participar en el futuro, principalmente en el resto de Europa.

Por ello, además de utilizar otras tecnologías estandarizadas que trabajan en bandas de frecuencia ISM, como Wi-Fi o Bluetooth, también cabe la posibilidad de utilizar tecnologías con protocolos propios no estandarizados. Es el caso de algunos sistemas de radio como los nRF24, que también trabajan en la banda de frecuencia de los 2.4 GHz, como por ejemplo los módulos nRF24L01.

El nRF24L01 [11] es un transceptor de radio de baja potencia que trabaja en la banda de los 2.4 - 2.4835 GHz, es compatible con Arduino y Raspberry Pi y es bastante sencillo de configurar e integrar. Utiliza un mecanismo de corrección de errores con un campo de control y otro de control de redundancia cíclica o *Cyclic Redundancy Check* (CRC), utiliza modulación *Gaussian Frequency Shift Keying* (GFSK) y es configurable a través de una interfaz SPI. Algunas de las configuraciones que permite son el canal de frecuencia

que utiliza (admite 6 canales) o la tasa de envío, que puede ser de 250 kbps, 1 Mbps o 2 Mbps.

A continuación, se muestra en la Fig. 3.11 el contenido de un paquete transmitido por el módulo nRF24L01.

- El campo *Preamble* o preámbulo puede tomar los valores 01010101 ó 10101010, dependiendo de si el primer bit del campo *Address* es un 1 o un 0. Se utiliza para asegurar que hay suficientes transiciones en el preámbulo para sincronizar el demodulador del receptor.
- *Address* o dirección es utilizado para indicar al receptor desde que emisor ha sido enviado ese paquete y evitar que interprete paquetes de otros sistemas nRF24L01.
- El campo *Packet Control Field* o campo de control de paquete está compuesto por tres subcampos internos como muestra la Fig. 3.12.
- *Payload* es el campo que contiene los datos que queremos enviar. En el caso del nRF24L01 es un campo configurable. Su longitud puede ir desde 0 a 32 bytes como máximo.
- Y por último, CRC es el mecanismo de detección de errores del paquete. Se calcula a partir de los campos *Address*, *Packet Control Field* y *Payload*. El receptor realiza el mismo cálculo sobre el paquete que recibe y en caso de que el resultado no sea el mismo que el valor contenido en este campo lo descarta, ya que esto significa que se ha recibido el paquete con algún error.
- Con el campo *Payload Length*, o longitud del payload, se indica el tamaño configurado para el payload. Como se ha comentado antes el payload puede tener longitudes de entre 0 y 32 bytes.
- *PID* es el campo utilizado para saber si un paquete es nuevo o es retransmitido. Es incrementado por el transmisor cada vez que genera un paquete nuevo. En el caso de que tenga que retransmitir un paquete, el valor de este campo será el mismo que el que tenía el paquete original a retransmitir. Este mecanismo también sirve para que el receptor no procese más de una vez el mismo contenido.
- El campo *NO_ACK* es un *flag* o marcador que indica el modo de retransmisión que se ha configurado. Si este bit es un 1 el emisor únicamente enviará paquetes sin importarle si han llegado o no, mientras que si este bit es un 0 por cada paquete transmitido deberá recibir un mensaje de confirmación, o *ACK*, por parte del receptor. En caso de no recibirlo procederá a retransmitir dicho paquete hasta que el receptor le confirme que lo ha recibido.

Para terminar, explicaremos en qué consiste la modulación GFSK que utiliza el transceptor nRF24L01. Se trata de una variación de la modulación *Frequency Shift Keying* (FSK), la cual consiste en modificar la frecuencia de la señal portadora dependiendo de si el bit transmitido es un 0 o un 1, como muestra la Fig. 3.13.

En el caso de la modulación GFSK se realiza un paso antes de modular con la metodología FSK: se pasa la señal original por un filtro gaussiano, como el mostrado en la Fig. 3.14, para suavizar la transición entre los símbolos 0 y 1 y así conseguir una señal portadora más limpia y fácil de demodular.

Decisión tomada

Tras el análisis realizado, las principales opciones son utilizar el estándar 802.11.p de Wi-Fi, ZigBee o un módulo nRF24, que utiliza un protocolo propio que trabaja en una frecuencia dentro de las bandas ISM.

El motivo que nos ha hecho decidir no utilizar ningún estándar Wi-Fi son los resultados del estudio plasmado en [12], en el cual podemos observar los resultados tras realizar varios experimentos utilizando los estándares 802.11a y 802.11p. En el primer caso la comunicación se pierde a partir de que el vehículo se desplace a una velocidad mayor que 20 km/h, mientras que al utilizar el estándar 802.11p con distintas modulaciones, en el mejor de los casos teóricos el alcance podría llegar a los 425 metros aproximadamente, mientras que en el caso práctico no consiguen establecer una comunicación a más de 220 metros aproximadamente.

En el caso de ZigBee, el punto de inflexión que nos ha hecho descartar esta opción ha sido también el alcance que ofrece, ya que comparado con otras soluciones es menor y demasiado bajo para nuestros requerimientos de poder cubrir un circuito de nuestra categoría, y los módulos XBee que pueden llegar a tener este alcance aumentan su precio considerablemente.

Por tanto, finalmente se ha decidido utilizar el transceptor nRF24L01. Nos ha parecido la mejor opción teniendo en cuenta que, aunque no llega al primer objetivo que nos planteamos de tener un alcance de 1000 metros, sí que podría llegar a los 600 metros, distancia más que suficiente para cubrir la longitud de cualquier circuito de *Formula Student*. Además es compatible con la plataforma Arduino, tiene un tamaño reducido y su precio bastante económico.

Por todos estos motivos y por su fácil integración con la plataforma Arduino que vamos a utilizar, es la opción elegida por considerarla la solución más sencilla y funcional de entre todas las contempladas.

3.4. Elección de dispositivos

Una vez tomada la decisión de utilizar la plataforma Arduino y los transceptores para el desarrollo de este sistema de telemetría, quedan por determinar qué placa de entre todas las posibilidades que ofrece Arduino se va a utilizar.

Microcontrolador

Arduino Uno R3

La placa Arduino Uno, mostrada en la Fig. 3.15 es la más básica de todas las de la plataforma Arduino.

Se trata de un microcontrolador basado en el microcontrolador ATmega328P que cuenta con 14 pines digitales, de los cuales 6 pueden ser usados como salida de señales PWM (pulse-width modulation), 6 pines analógicos de entrada, un reloj electrónico de cuarzo de 16MHz, conector a USB y 32kB de memoria flash.

Es bastante fácil de programar, ya que cuenta con un adaptador AC-to-DC que permite que se pueda conectar al ordenador mediante el conector USB sin problema.

Para escribir su código, al igual que todas las placas Arduino, cuenta con el software IDE de Arduino para facilitar su programación.

A pesar de ser una de las placas más básicas de Arduino es una buena opción para desarrollar pequeños proyectos debido a su pequeño tamaño y reducido peso. Se trata de un microcontrolador que mide únicamente 68.6 mm de longitud y 53.3 mm de anchura, y sólo pesa 25 gramos.

Arduino Mega

Como se puede observar en la Fig. 3.16, la placa Arduino Mega es la evolución de la Arduino Uno, contando en esta ocasión con el microcontrolador ATmega2560. Además de con todas las características de la placa Arduino Uno también cuenta con 54 pines digitales, de los cuales hasta 15 pueden ser utilizados como salida de señales PWM, 16 entradas analógicas, 256kB de memoria flash y 4 puertos hardware UART para comunicación serial, mientras que la placa Arduino Uno solo cuenta con un puerto UART.

Así, el microcontrolador Arduino Mega cuenta con un mayor número de pines digitales y analógicos y con mayor memoria flash, aunque mantiene la velocidad de su reloj digital en 16MHz.

Este modelo es la placa más potente con microcontrolador de 8 bits y el que más pines tiene con respecto al resto. A cambio, su precio es un poco más elevado y como es de esperar su tamaño también es mayor. Concretamente mide 101.52 mm de longitud y tiene una anchura de 53.3 mm. Y su peso es de 37 gramos.

Arduino Nano

Arduino Nano es una placa compacta con especificaciones muy parecidas a las de la placa Arduino Uno. Este modelo de placa se muestra en la Fig. 3.17.

Al igual que la placa Arduino Uno, está basada en el microcontrolador ATmega328P, cuenta con 32kB de memoria flash y un reloj electrónico de cuarzo de 16MHz. Como principales diferencias, en el caso del Arduino Nano se cuenta con un puerto mini-B USB en lugar del estándar y cuenta con un mayor número de pines: 22 pines digitales, 6 de los cuales pueden ser usados como salida de señales PWM y 8 pines analógicos.

Otra de sus ventajas es su reducido tamaño y peso, ya que únicamente mide 45 mm de largo y 18 mm de ancho y sólo pesa 7 gramos.

Módulos de radio

Módulos nRF24L01

Estos transceptores, analizados en el apartado 3.3, ofrecen dos variantes: los módulos con antena integrada y los módulos con antena externa. En el primer caso, el alcance teórico apenas llega a los 100 metros, mientras que el módulo con antena externa ofrece un rango de aproximadamente 600-700 metros. Por este motivo, hemos decidido utilizar el módulo con antena externa de la Fig. 3.18.

Los módulos nRF24L01 se alimentan a 3.3 voltios, y aunque las placas de Arduino ofrecen una salida de 3.3V, se recomienda utilizar un regulador de tensión para alimentar con una corriente más estable a estos módulos. Por ello, más adelante se comprobará si realmente afecta a su rendimiento para utilizarlo en nuestro sistema o no.

Se comunican con la placa de Arduino mediante comunicación SPI y cuenta con una librería para el IDE de Arduino que simplifica mucho la programación de sus funcionalidades básicas.

Decisión tomada

Tras comparar las características de las tres placas aquí descritas hemos tomado la decisión de utilizar un microcontrolador Arduino Mega para colocarlo dentro del monoplaza. A pesar de ocupar un mayor espacio, es la única placa con pines suficientes para poder conectar todos nuestros sensores. Además, es un poco más potente, lo que posiblemente nos permita realizar un mayor número de transmisiones que otras placas Arduino en el mismo espacio de tiempo.

En cuanto al módulo de radio, como se argumentó anteriormente, hemos decidido utilizar el módulo nRF24L01 con antena externa frente a las demás opciones por su bajo precio, la sencillez que ofrece su integración con Arduino y su rango de emisión. Para estar seguros de que cumple con nuestros requisitos más adelante se realizarán pruebas

para comprobar su alcance real.

Al ser una función en la que no nos importa demasiado el tamaño o el peso, para la función del receptor nos da igual qué placa utilizar. Utilizaremos una placa Arduino Uno por sencillez.

Para alimentar a la placa Arduino Mega se ha extraído la energía necesaria de la batería del coche. Pero se ha colocado un regulador de corriente continua de 20V a 8V para evitar posibles sobretensiones, ya que, según sus especificaciones, las cuales se pueden encontrar en la página oficial de Arduino [13], no se debe alimentar a la placa con más de 20V, cifra que podría llegar a alcanzar la batería del coche.

Periféricos

Módulo SD card

Una vez que ya conocemos la placa de Arduino que vamos a utilizar para recopilar todos los datos de los sensores antes de transmitirlos ya podemos buscar soluciones para determinar cómo vamos a realizar la fase de *datalogger* para almacenar todos los datos transmitidos.

Es necesaria esta fase ya que no todos los datos de los sensores recopilados serán enviados por la comunicación radio, como es el caso de los amortiguadores, que interesan más para su posterior análisis ya que en tiempo real solo se verían multitud de variaciones incomprensibles. Además, también es importante contar con una copia de seguridad o *backup* para analizar con posterioridad cuándo y por qué se han producido cambios significativos en las medidas de los sensores.

Dado que se va a utilizar la placa Arduino Mega se ha buscado un periférico de Arduino compatible que permita almacenar toda la información. De entre todas las posibilidades se ha determinado que el módulo para tarjetas SD de Arduino es la mejor opción para implementar la fase de *datalogger*, la cual se muestra en la Fig. 3.19.

Arduino Mega Sensor Shield

Este módulo o *Shield* para Arduino de la Fig. 3.20 facilita realizar las conexiones con los pines de la placa Arduino Mega. Debido a que la placa irá colocada en el coche dentro de una caja fabricada mediante impresión 3D, y al gran número de sensores que irán conectados a ella, tomamos la decisión de colocar este módulo para hacer más sencilla la conexión de todos los cables correspondientes a cada sensor, del módulo para la tarjeta SD y del emisor que transmitirá todos los datos de nuestro sistema de telemetría.

3.5. Soportes para el sistema

3.5.1. Diseño

Para colocar los distintos módulos del sistema de telemetría en el coche, se han diseñado diferentes cajas para ser impresas en plástico ABS mediante impresión 3D. Este trabajo fue realizado con la ayuda de un compañero del equipo que conoce mejor la herramienta software de diseño *SolidWorks*.

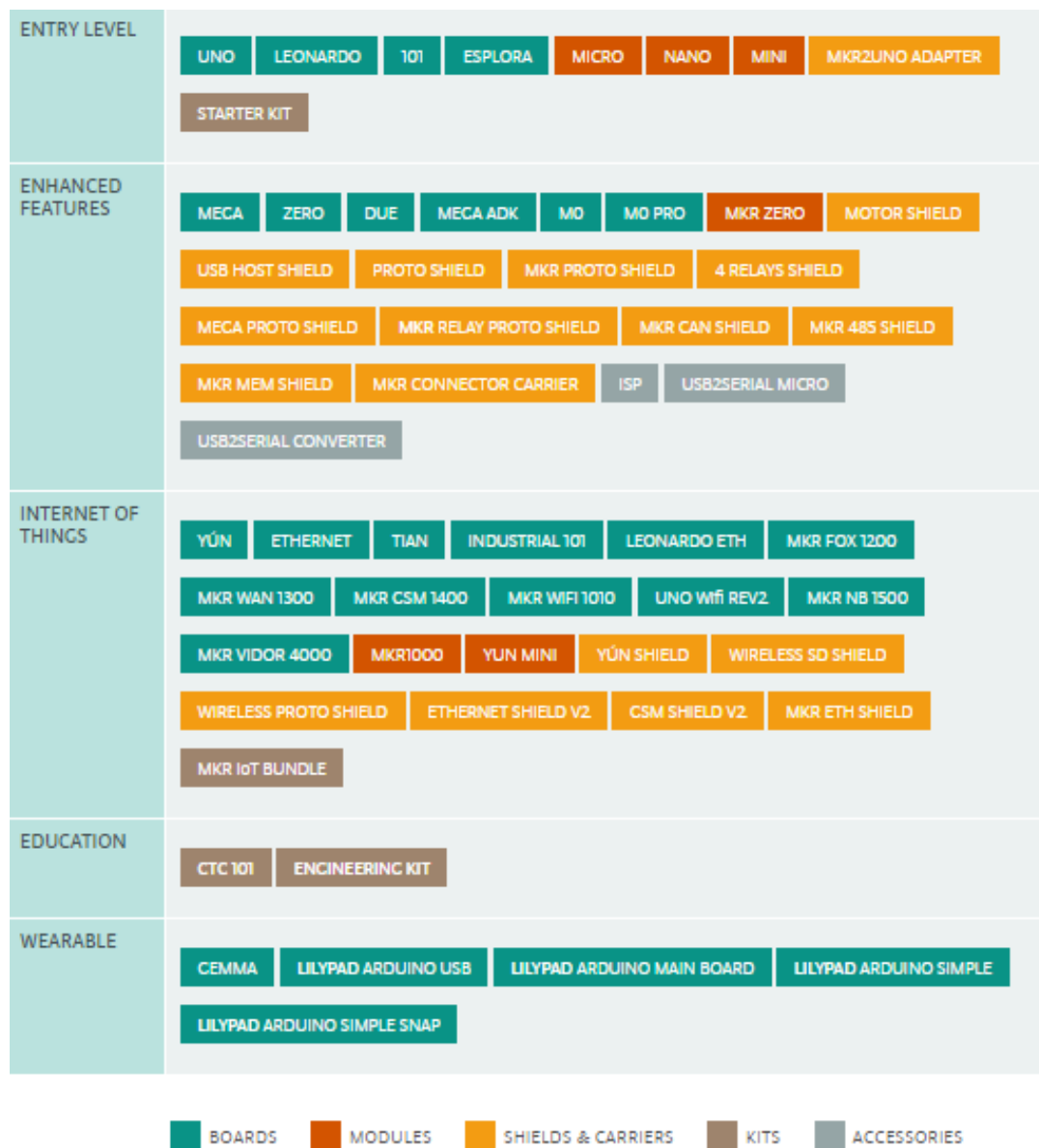
Se han diseñado e impreso tres cajas diferentes: una para albergar la placa Arduino Mega, a la cual se conectarán todos los cables de los sensores; otra para el *Shield* de la tarjeta SD; y por último una cubierta con un soporte al que ajustarse para el módulo emisor nRF24L01. Dichas cajas son las que se pueden ver en la Fig. 4.1.

3.5.2. Ubicación

Los módulos de recepción de los datos de todos los sensores y del *datalogger*, es decir, las cajas que albergan al microcontrolador Arduino y al módulo de la tarjeta SD, irán colocadas junto con el resto de electrónica del coche. Aprovechando la forma del monocasco o sub-chasis delantero de este año, todas las cajas de electrónica irán colocadas en una de sus columnas, más concretamente en la columna derecha.

En cuanto al módulo emisor de radio, irá colocado en uno de los brazos laterales del sub-chasis tubular trasero, tan alto como permita la comunicación SPI, y evitando cualquier posible interferencia con elementos del coche.

Dichos módulos se muestran en su ubicación dentro del coche en la Fig. 3.21.



[Imagen]. Recuperado de <https://www.arduino.cc/en/Main/Products>

Fig. 3.9. Variedad de placas Arduino



[Imagen]. Recuperado de <https://www.mouser.es/new/digi-international/digi-xbee-s1-rf-modules/>

Fig. 3.10. Módulos XBee



Fig. 3.11. Paquete transmitido por el nRF24L01 [11]



Fig. 3.12. Campo de control de paquetes (*Packet Control Field*) [11]

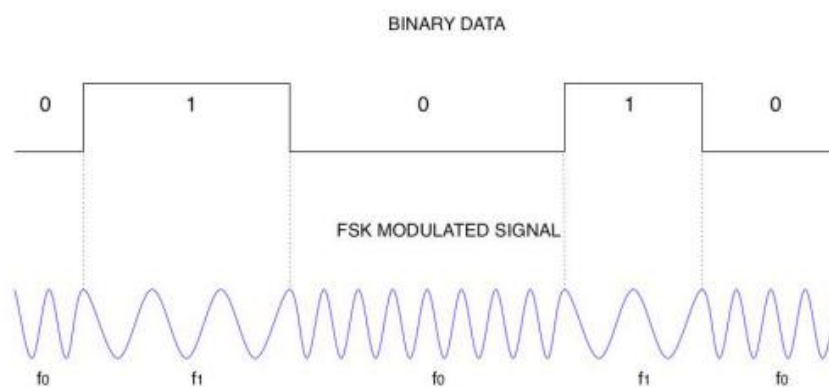
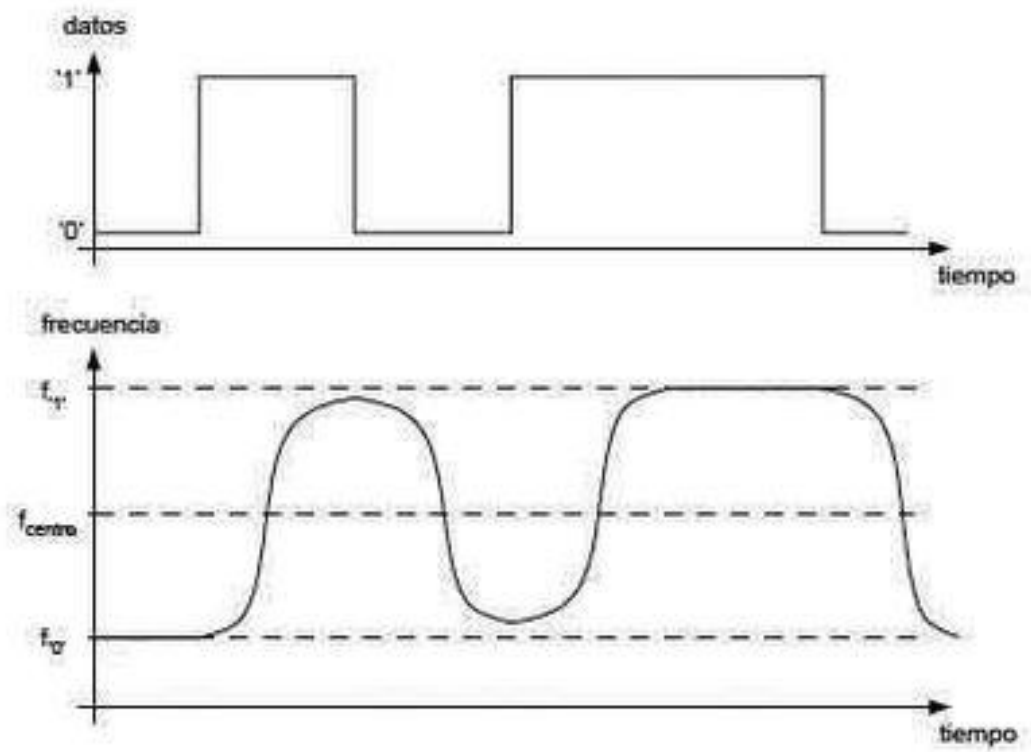


Fig. 3.13. Modulación FSK



[Imagen]. Recuperado de

https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_desplazamiento_de_frecuencia_gausiana

Fig. 3.14. Filtro Gaussiano

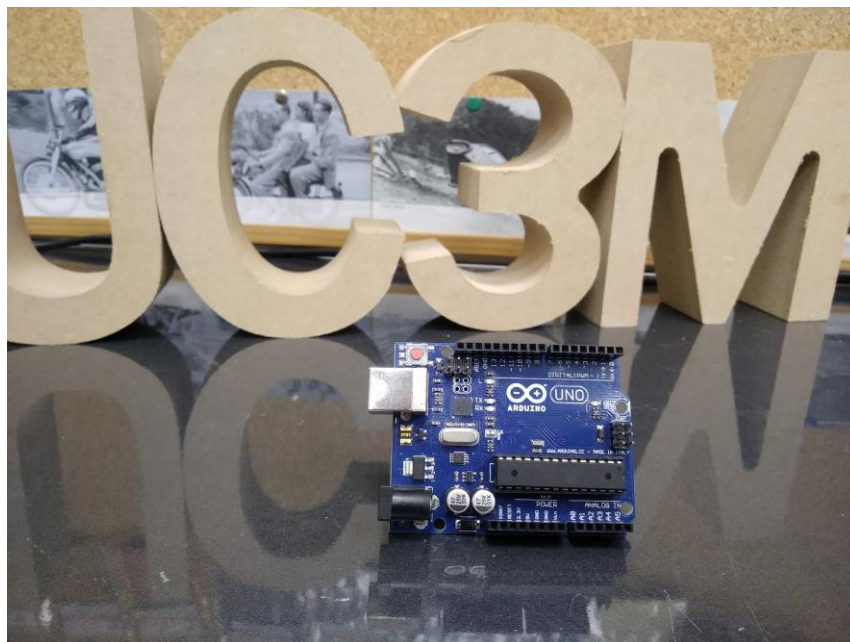


Fig. 3.15. Placa Arduino Uno



Fig. 3.16. Placa Arduino Mega

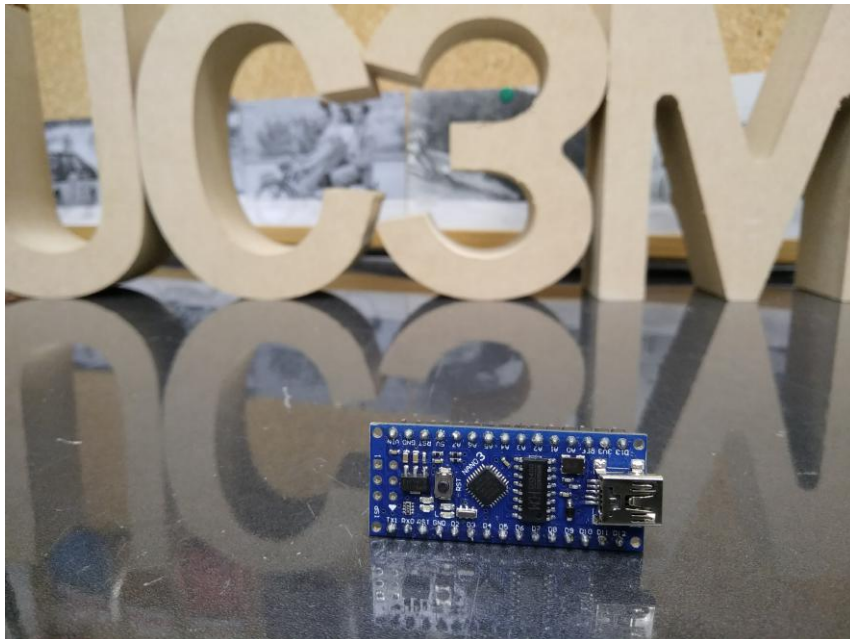


Fig. 3.17. Placa Arduino Nano



Fig. 3.18. Módulo nRF24L01

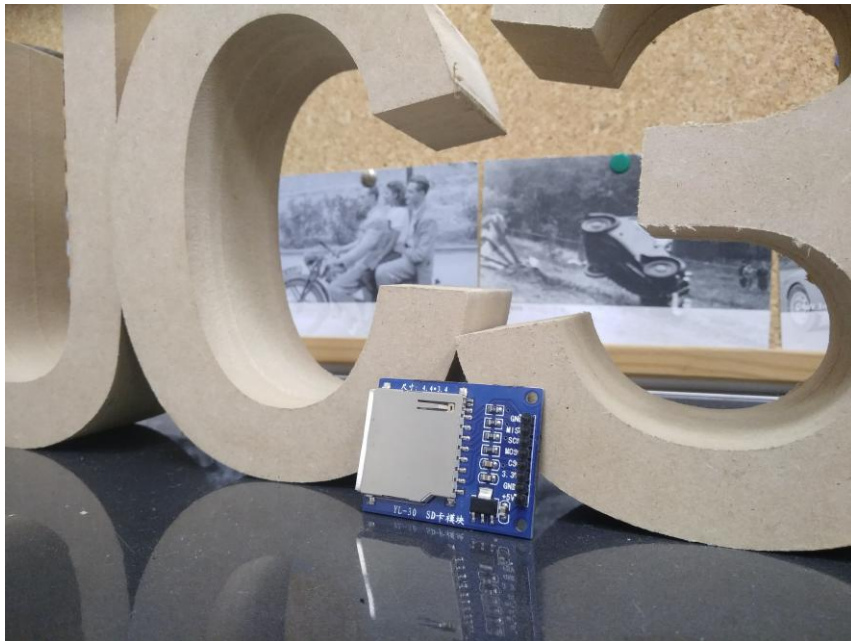


Fig. 3.19. Módulo tarjeta SD

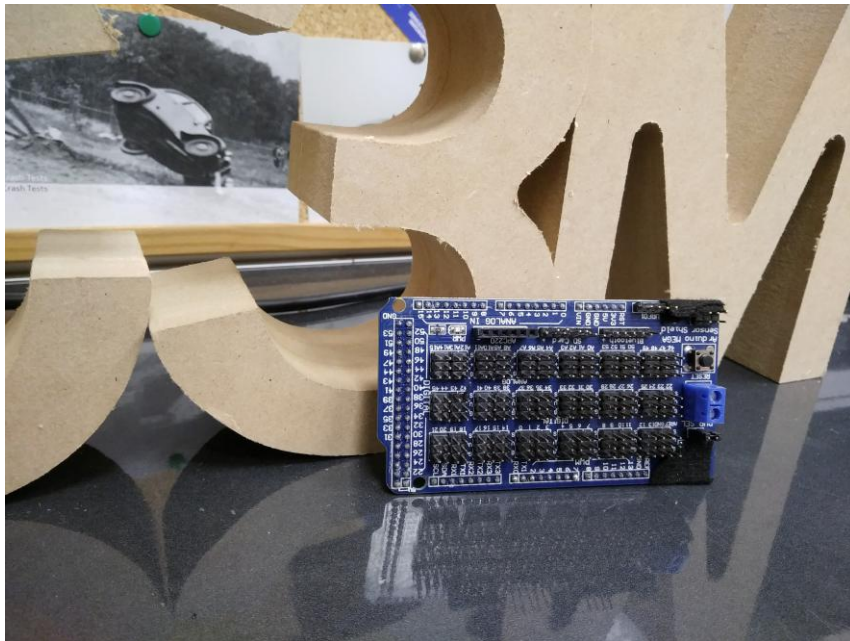


Fig. 3.20. Arduino Mega Sensor Shield



Fig. 3.21. Primer escenario de la prueba de alcance

4. DESARROLLO DEL SISTEMA

Una vez conocidos todos los elementos de los que va a constar nuestro sistema de telemetría, ya podemos comenzar con su desarrollo.

Para empezar se han realizado todas las conexiones entre los distintos dispositivos. Este montaje se ha dividido en varios módulos pensados para poder conectarlos y desconectarlos de manera fácil y rápida, sin contar con todos los sensores, que se quedarán colocados en el coche preparados con conectores para poder conectarse mediante dos mangueras de cables al Arduino Mega, una para la parte delantera del vehículo y otra para la trasera.

En total el sistema estará formado por cuatro módulos diferentes:

- El primero y principal estará compuesto por la placa Arduino Mega y el *Sensor Shield*, encajados perfectamente uno encima del otro.
- El segundo será el módulo nRF24L01 encargado de emitir los datos recopilados.
- El tercero será el módulo *datalogger*, compuesto por el *Shield* de la tarjeta SD, que guardará todos los datos para proceder a su posterior análisis.
- Y por último, el módulo receptor, compuesto por la placa Arduino Uno y el transceptor nRF24L01, que recibirá los datos enviados por el emisor desde el coche.

Además, se contará con todos los sensores repartidos por el coche y dos mangueras de cables que conectarán estos sensores con el módulo principal.

En la Fig. 4.1 se muestran los tres módulos que irán colocados en el coche.

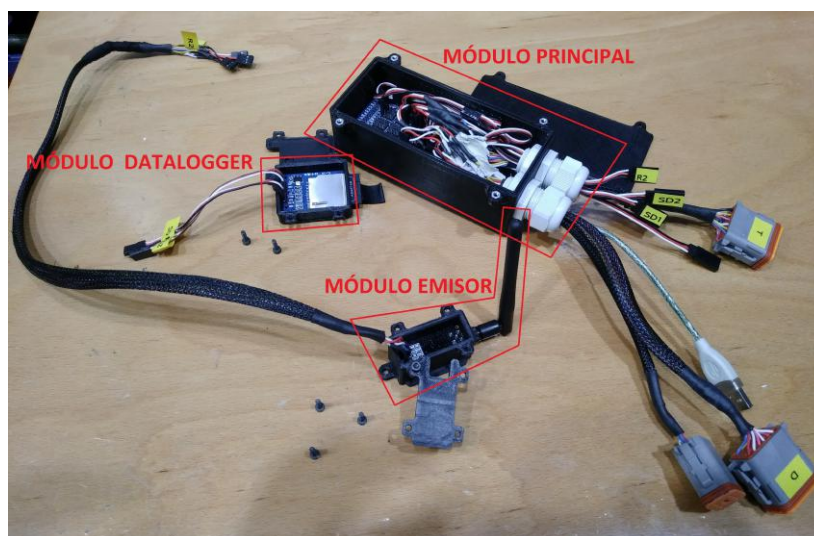


Fig. 4.1. Módulos del sistema colocados en el coche

Y A continuación se presenta un esquema del montaje en la Fig. 4.2 para aclarar la disposición de cada módulo dentro del coche.

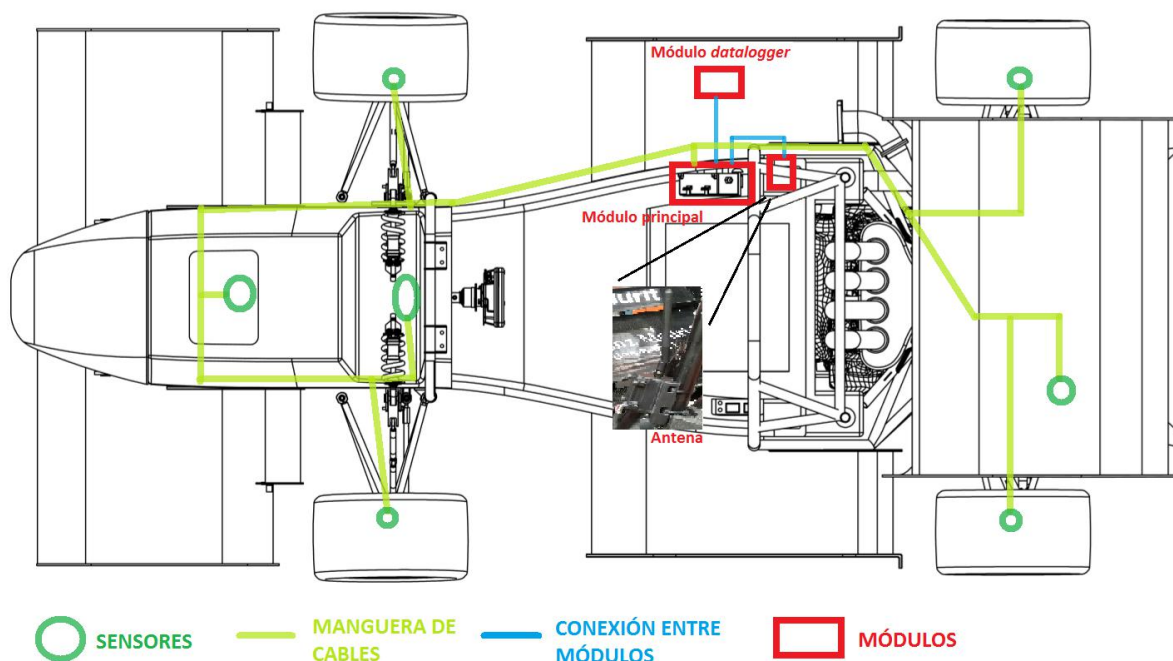


Fig. 4.2. Esquema del montaje en el coche

4.1. Programación del emisor

Una vez ya están todos los módulos conectados o preparados para conectarse entre sí, es necesario desarrollar el software para conseguir que cada módulo cumpla su función. Para ello, se han creado dos programas distintos utilizando el IDE de Arduino.

El primero de ellos es el cargado en la placa Arduino Mega. Este programa está diseñado para que el microcontrolador reciba todos los datos de los sensores, los trate y transforme en datos comprensibles para nosotros en los casos que sea necesario, y los comunique tanto a la tarjeta SD para guardarlos como al módulo nRF24L01 para que los transmita.

En la Fig. 4.4 se muestra el diagrama de flujo del programa empleado en este módulo, y el código completo en el Anexo B.

4.2. Programación del receptor

En este caso se ha cargado el código presente en el Anexo B en la placa Arduino Uno del módulo de recepción.

Este programa conecta el módulo nRF24L01 al mismo canal de comunicación que utiliza el módulo emisor. Una vez conectados, el receptor espera hasta encontrar bytes

que poder leer en el canal. Cuando recibe información, la recoge y envía por el puerto serial al programa encargado de la visualización de estos datos en el formato adecuado.

El diagrama de flujo de este programa se encuentra en la Fig. 4.5.

4.3. Interfaz de visualización

Para visualizar los datos se ha utilizado una herramienta llamada *Telemetry Viewer*, en su versión 0.4 [14].

Se trata de un programa que permite conectarse al puerto serial de la placa Arduino que se esté utilizando y configurar cómo y qué datos de los que recibe por este puerto van a mostrarse.

Los datos seleccionados pueden dividirse y dibujarse en distintas gráficas que pueden visualizarse simultáneamente. Estas gráficas van añadiendo los datos recibidos por el puerto serial en tiempo real, por lo que son gráficas dinámicas que se pueden utilizar para monitorizar el estado de los sensores en todo momento. Como ejemplo se muestra la Fig. 4.3.

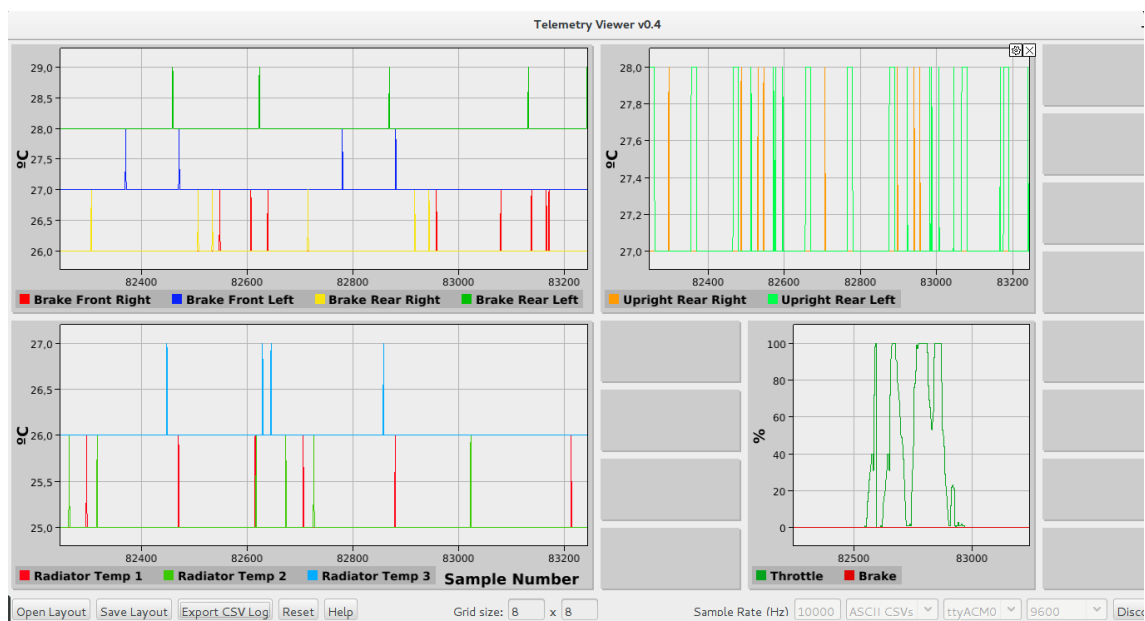


Fig. 4.3. Captura de pantalla de la herramienta Team Viewer

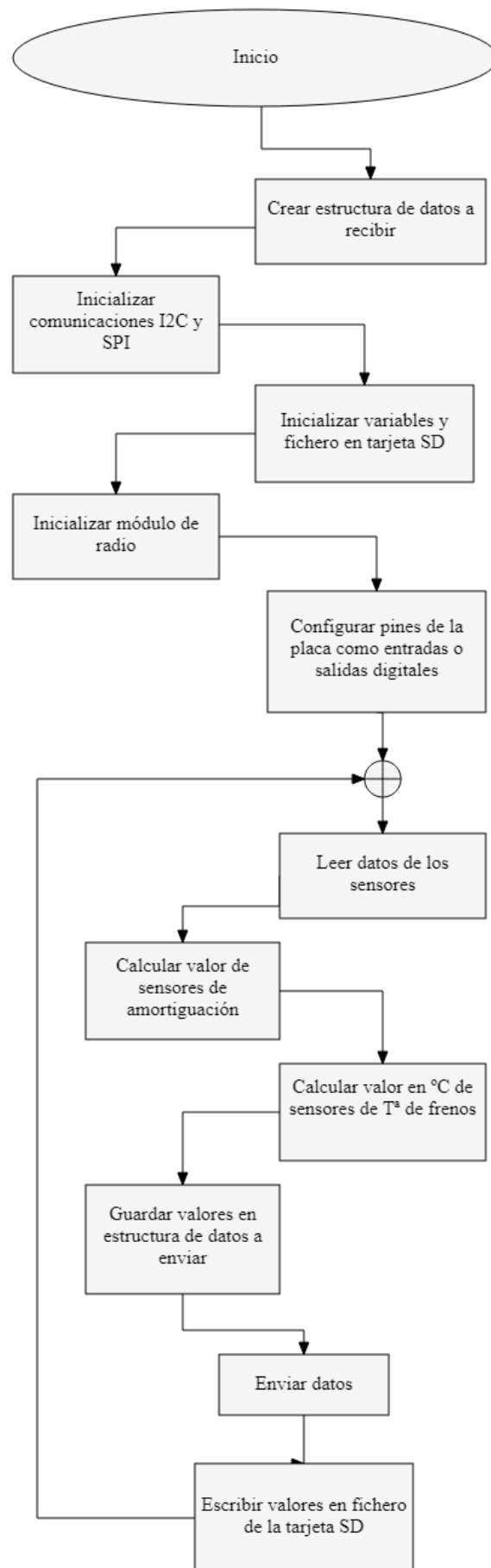


Fig. 4.4. Diagrama de flujo del código del módulo emisor

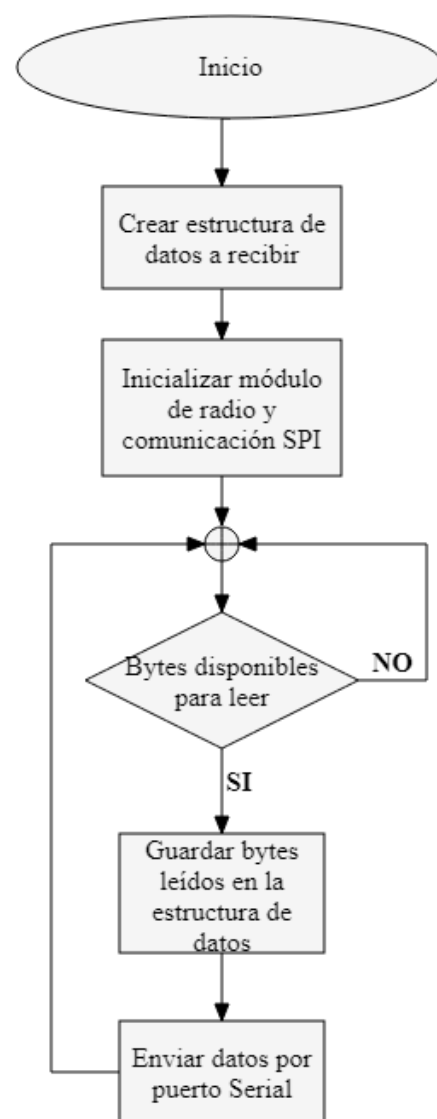


Fig. 4.5. Diagrama de flujo del código del módulo receptor

5. PRUEBAS REALIZADAS

Para comprobar y tomar como definitiva la decisión de utilizar los módulos de radio nRF24L01 como dispositivos de emisión y recepción de los datos que queremos enviar se realizaron varias pruebas, que se describirán a continuación.

5.1. Pruebas de los módulos de radio

Antes de completar la fase de preparación de todos los módulos para su interconexión, cuando únicamente se había probado con un programa sencillo que los dos módulos nRF24L01 funcionaban y se comunicaban entre sí, un compañero del equipo nos ofreció un par de antenas más largas (a partir de ahora antenas B) que las originales de los transceptores (a partir de ahora antenas A). Eran antenas que también operan en la banda de los 2.4 GHz, por lo que muy probablemente serían compatibles.

Tras colocarlas en los módulos de radio y comprobar que también nos podían servir, debíamos tomar la decisión de qué antenas utilizar entre las mostradas en la Fig. 5.1 para nuestro sistema: ¿las antenas B o las antenas A, a pesar de ser más cortas?



Fig. 5.1. Antenas A (11 cm) y B (21 cm)

5.1.1. Prueba de diferentes antenas

Como no conocíamos nada acerca de las antenas B, para caracterizarlas y así poder compararlas con las antenas A, pedimos ayuda al profesor titular de la Universidad Luis de Inclán Sánchez, que muy amablemente nos ayudó con nuestro propósito.

Luis nos ofreció el acceso a un laboratorio del departamento de Teoría de la Señal y Comunicaciones (TSC) y a sus aparatos para poder medir algunos parámetros de nuestras antenas con el objetivo de compararlas y poder tomar una decisión basada en experimentos.

Descripción

Para realizar la caracterización de las antenas realizamos dos pruebas diferentes utilizando los dispositivos que se citarán a continuación, los cuales se pueden ver en las figuras Fig. 5.2, Fig. 5.3 y Fig. 5.4 :

Analizador de red o *Vector Network Analyzer* Modelo: HP Agilent 8753E

Utilizado para medir los parámetros S, o parámetros de dispersión, que más adelante se explicarán, y comprobar cómo de eficientes son las antenas en un rango de frecuencias determinado.



Fig. 5.2. Analizador de red HP Agilent 8753E

Generador de señales Modelo: HP Agilent N5181A

Utilizado para generar una señal a una frecuencia de 2.4 GHz.



Fig. 5.3. Generador de señales HP Agilent N5181A

Antena multibanda directiva

Colocada para transmitir la señal recibida desde el generador de señales en dirección a las antenas a caracterizar.

Analizador de espectros *Modelo: HP 8594E*

El analizador de espectros ha sido utilizado para medir la potencia recibida con cada una de las antenas de la señal emitida por la antena directiva,

El objetivo de la primera prueba era conocer el funcionamiento de cada una de las antenas a la frecuencia en la que nuestros dispositivos trabajarán (2.4 GHz). Es decir, comprobar para qué bandas de frecuencia la antena está optimizada, y más concretamente, cuántas pérdidas introduce la antena a esa frecuencia. Para ello, medimos el parámetro S_{11} , más conocido como coeficiente de reflexión.

Los parámetros S describen la relación de entrada y salida entre puertos en un sistema eléctrico. En general, el parámetro S_{NM} representa la potencia transmitida desde el puerto M hasta el puerto N . Así, si contamos con dos puertos, el parámetro S_{12} representa la potencia transferida desde el puerto 2 hasta el puerto 1.

En la práctica, el parámetro más utilizado referido a antenas es el S_{11} . Este parámetro representa cuánta potencia es reflejada por la antena, por eso es conocido como coeficiente de reflexión. Si S_{11} es 0 dB, significa que toda la potencia es reflejada por la antena y no se radia nada. Así, si S_{11} es -10 dB, esto implica que si se le entrega a la antena una potencia de 3 dB, -7 dB es la potencia reflejada.



Fig. 5.4. Analizador de espectros HP 8594E

Para la segunda prueba, nuestro objetivo era comprobar el comportamiento de las antenas en recepción. Para ello, construimos un sistema donde transmitiendo con una misma antena, probamos las antenas A y B como receptoras para comprobar cuál introduce más o menos pérdidas. En la Fig. 5.5 se puede observar el escenario empleado.



Fig. 5.5. Prueba realizada en el laboratorio

La antena utilizada para la transmisión ha sido la antena multibanda directiva mencionada anteriormente. Mediante el generador de señales se generaba una señal a 2.4 GHz y con el analizador de espectros se medía la potencia recibida con cada una de las antenas.

Resultados

Realizamos ambas pruebas dos veces con cada tipo de antena, una primera vez con las antenas rectas y una segunda vez con las antenas dobladas, como se muestra en las imágenes de la Fig. 5.6.

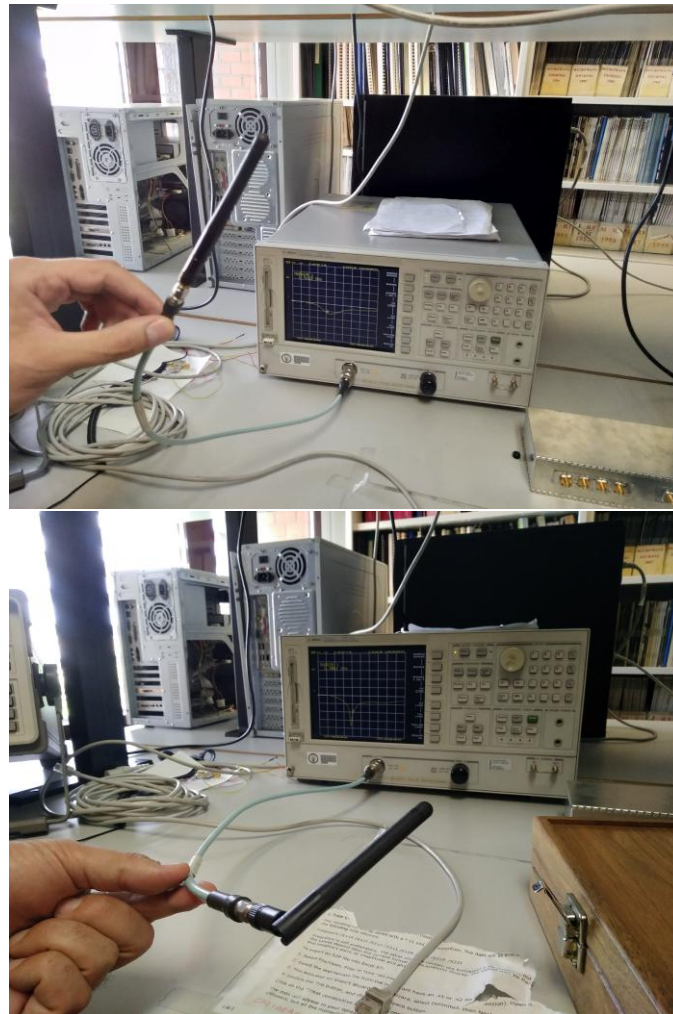


Fig. 5.6. Posición de las antenas para la primera prueba realizada en el laboratorio

La Tabla 5.1 presenta los rangos de frecuencia en los que el parámetro S_{11} alcanzaba un valor de -10 dBm durante la realización de la primera prueba:

	Antena recta	Antena doblada
Antena A	De 2.35 GHz a 2.57 GHz	De 2.30 GHz a 2.48 GHz
Antena B	De 2.61 GHz a 2.70 GHz	De 2.64 GHz a 2.71 GHz

Tabla 5.1. Resultados prueba medición del parámetro S_{11}

Con estos resultados podemos llegar a la conclusión de que las antenas A están mejor adaptadas para trabajar a 2.4 GHz, que es la frecuencia a la que teóricamente emiten los módulos nRF24L01. Además se puede concluir que, aunque no hay mucha diferencia, las antenas trabajan mejor a esa frecuencia estando rectas, en lugar de dobladas.

Para la segunda prueba, en la que medimos la potencia recibida por cada una de las antenas con el medidor de espectros, obtuvimos los resultados mostrados en la Tabla 5.2.

	Potencia recibida
Antena A	-56 dBm
Antena B	-61 dBm

Tabla 5.2. Resultados prueba potencia recibida

Estos resultados coinciden con los obtenidos durante la primera prueba en que las antenas A son la mejor opción para operar en una comunicación a 2.4 GHz.

5.1.2. Prueba de alcance

Descripción

Para comprobar la distancia real a la que es posible una comunicación entre dos módulos nRF24L01 se ha llevado a cabo esta prueba de alcance. Para llegar a una conclusión con sus resultados existe la posibilidad de analizar dos parámetros distintos, el *Bit Error Rate* (BER) y el *Frame Loss Ratio* (FLR).

El BER mide la cantidad de bits erróneos que se han recibido con respecto al número total de bits recibidos, y se expresa de la siguiente manera:

$$BER = \frac{N^{\circ} \text{ Bits Erróneos Recibidos}}{N^{\circ} \text{ Total Bits Recibidos}}$$

Pero debido a que los módulos de radio nRF24L01 implementan un mecanismo de corrección de errores, no tiene sentido calcular este indicador, ya que el propio módulo descartará todos los paquetes que contengan algún error.

Por ello, nos hemos centrado en calcular el FLR. El FLR representa el ratio entre paquetes enviados y paquetes recibidos, calculado de la siguiente forma:

$$FLR = \frac{N^{\circ} \text{ Paquetes No Recibidos}}{N^{\circ} \text{ Total Paquetes Transmitidos}}$$

Para poder calcularlo, se decidió realizar una prueba en la que se enviarían 50 paquetes iguales de 32 bytes, ya que es el máximo payload que permite el estándar que utilizan los módulos nRF24L01. El contenido de estos paquetes eran 32 números, del 1 al 32, que se revisarían tras las pruebas para asegurarnos de que efectivamente estos módulos de radio

aplicaban un mecanismo de corrección de errores. El código Arduino empleado para esta prueba se muestra en el Anexo C.

La prueba consiste en medir a distintas distancias la cantidad de paquetes que se reciben, de entre los 50 enviados, y comprobar el alcance máximo que puede llegar a alcanzar la radiocomunicación.

Para determinar que la comunicación entre los dos módulos de radio ha sido correcta, establecimos que al menos se debe haber recibido la mitad de los paquetes transmitidos, es decir, un mínimo de 25 paquetes.

Resultados

Finalmente se ha realizado esta prueba en dos escenarios distintos. El primero de ellos ha sido en un descampado en la localidad de Getafe, el cual se muestra en la Fig. 5.7, donde la antena emisora se colocaba en el punto A y la receptora se iba alejando hasta el punto B. Se eligió este emplazamiento por ser una zona amplia en la que las antenas pueden mantener una visión directa entre ellas.

La primera prueba que se ha realizado ha sido la de comprobar el máximo alcance obtenido en la comunicación utilizando un regulador de tensión de 5 V a 3.3 V entre la placa de Arduino y el transceptor de radio, repitiendo la prueba alimentando al módulo de radio directamente desde el Arduino.

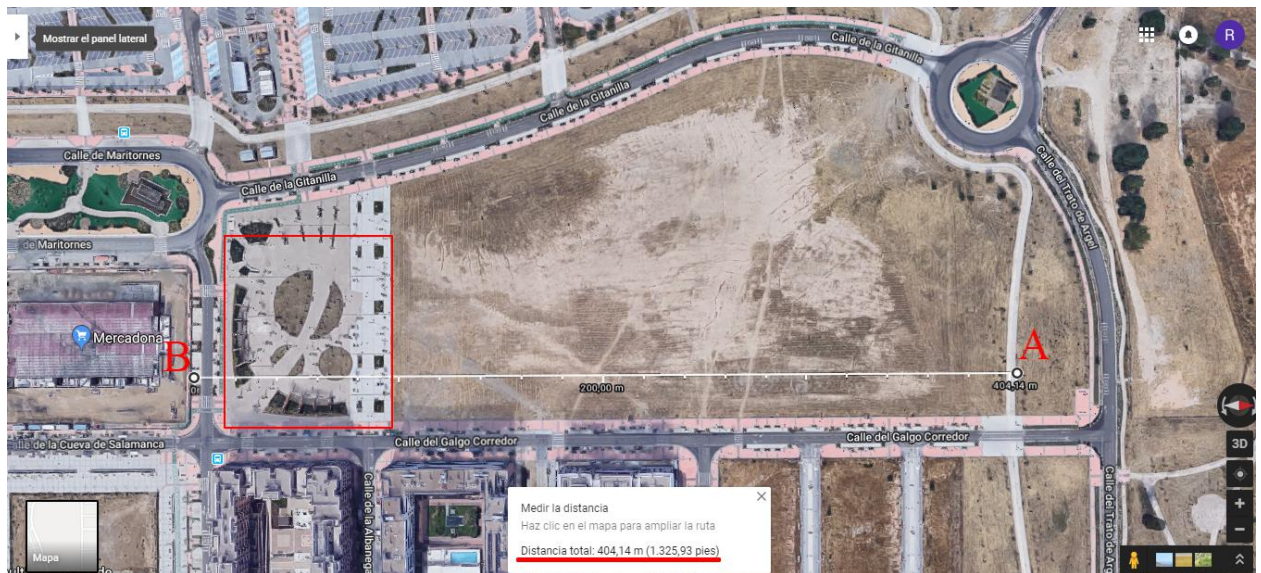


Fig. 5.7. Primer escenario de la prueba de alcance

En esta ocasión la prueba consistía en transmitir de manera ininterrumpida un mismo paquete e ir alejando poco a poco la antena receptora hasta que se empezaran a perder paquetes de manera considerable. Como comprobación, se tomó la decisión de que la comunicación se había perdido si pasaban más de cinco segundos sin recibir ningún paquete, los cuales se enviaban cada medio segundo.

Los resultados de esta prueba se muestran en la Tabla 5.3.

	Distancia sin regulador	Distancia con regulador
Antenas A	349 metros	390 metros
Antenas B	334 metros	387 metros

Tabla 5.3. Resultados prueba con y sin regulador

Claramente parece que utilizar el regulador de tensión afecta a la distancia que puede alcanzar la comunicación. Por este motivo, se ha tomado la decisión de utilizar de aquí en adelante el regulador de tensión entre la placa Arduino y el nRF24L01 tanto en el módulo emisor como en el receptor.

Además, con estos datos se observa que los resultados obtenidos en esta prueba y en el laboratorio coinciden, es decir, que las antenas A funcionan mejor que las B en nuestro caso. Pero no son datos suficientes como para tomar una decisión.

Por ello, se ha llevado a cabo una segunda prueba, concretamente la prueba de enviar 50 paquetes y comprobar cuántos de ellos se pierden.

Los resultados de dicha prueba, realizada en el primer escenario, se pueden observar en las tablas Tabla 5.4 y Tabla 5.5 y en las figuras Fig. 5.8 y Fig. 5.9.

	150m	200m	250m	300m	350m	375m	380m	385m	390m	395m	400m
Antenas A	50	50	50	50	48	39	48	49	35	36	31
Antenas B	49	48	49	50	49	40	46	45	15	0	0

Tabla 5.4. Número de paquetes recibidos a diferentes distancias
(escenario 1)

	150m	200m	250m	300m	350m	375m	380m	385m	390m	395m	400m
Antenas A	0	0	0	0	0.04	0.22	0.04	0.02	0.3	0.28	0.38
Antenas B	0.02	0.04	0.02	0	0.02	0.2	0.08	0.1	0.7	1	1

Tabla 5.5. FLR en el escenario 1

Como se puede observar, parece que los datos recogidos sustentan los resultados obtenidos en el laboratorio. Aun así, decidimos repetir la prueba en otro escenario por dos razones principales. La primera es que no pudimos medir a más de 400 metros de la antena emisora porque el recinto de un supermercado nos limitaba físicamente. Por este motivo no pudimos medir la máxima distancia a la que las antenas A podían mantener la comunicación.

Paquetes recibidos en escenario 1

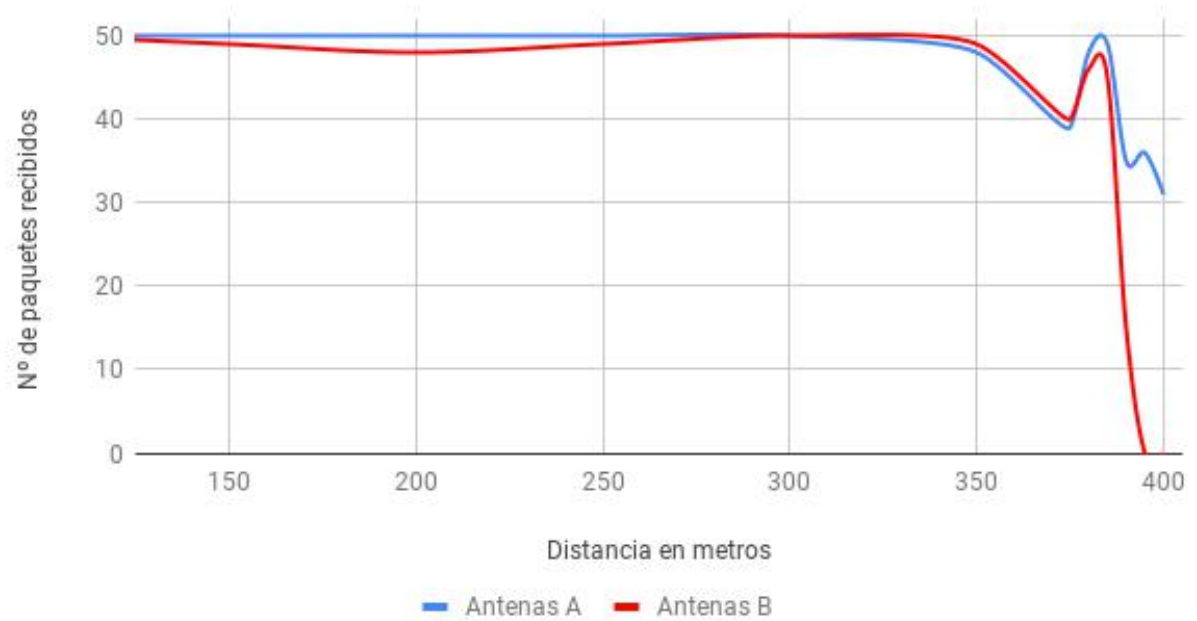


Fig. 5.8. Gráfica de número de paquetes recibidos a diferentes distancias (escenario 1)

FLR (Escenario 1)

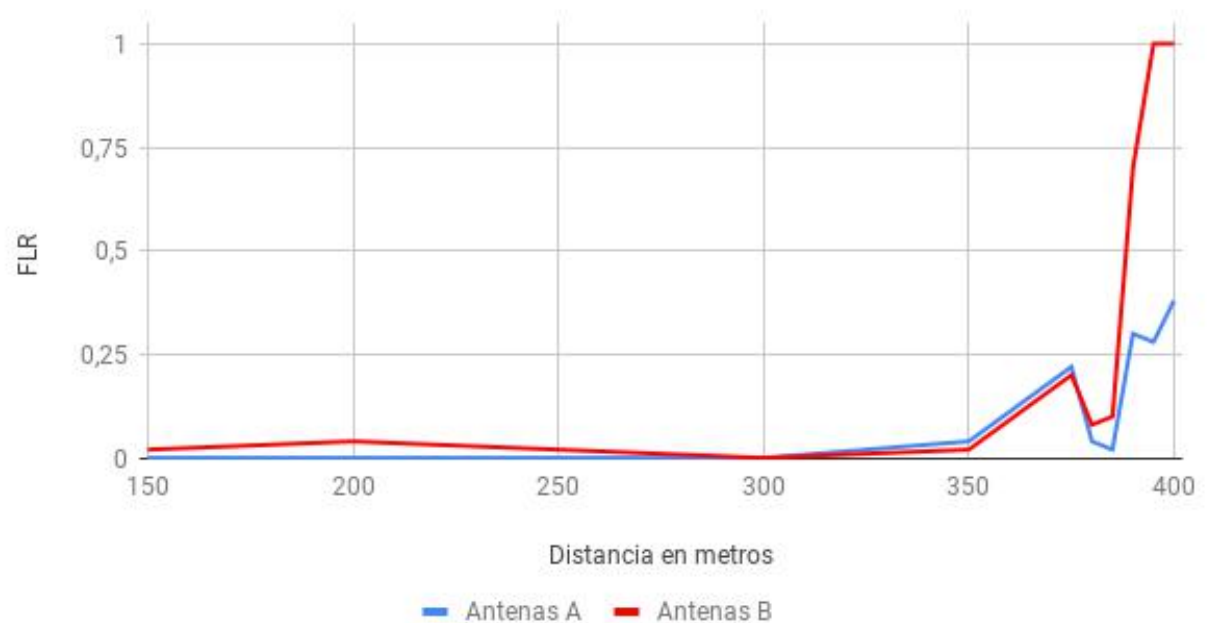


Fig. 5.9. Gráfica del FLR en el escenario 1

Y la segunda es porque, como se puede observar en la gráfica de la Fig. 5.8, entre los 350 y los 380 metros hay un valle en el que se dejaban de recibir paquetes. Esto puede

ser posible porque en la zona del recuadro rojo en la Fig. 5.7 hay un parque con algunos árboles pequeños que probablemente afectaran a la comunicación, aunque durante la realización de la prueba siempre se intentaron evitar para mantener una línea de visión directa entre las antenas.

Para la repetición de la prueba de alcance se eligió un escenario con suficiente distancia en línea recta y sin ningún tipo de obstáculo entre medias. El lugar elegido fue una calle del polígono Los Olivos, en la localidad de Getafe. Como se puede ver en la Fig. 5.10, se trata de una calle con una longitud de aproximadamente 640 metros, donde de nuevo la antena emisora estaba situada en el punto A y la receptora se iba alejando con dirección al punto B.

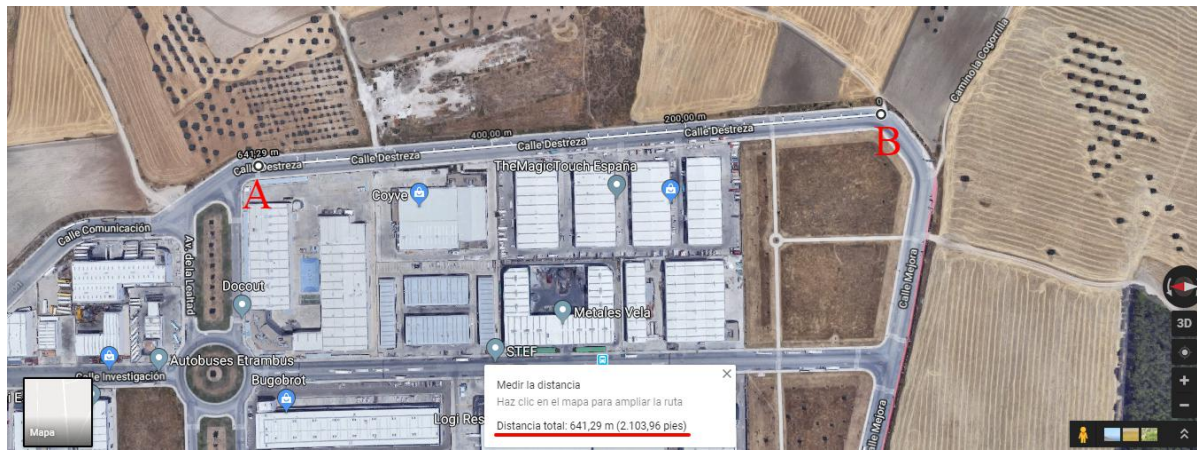


Fig. 5.10. Segundo escenario de la prueba de alcance

En este caso, para que los resultados fueran más fiables, se realizaron dos repeticiones de cada medida para finalmente calcular la media aritmética de los paquetes recibidos en cada punto. Los resultados obtenidos se representan en las tablas Tabla 5.6 y Tabla 5.7 y en las figuras Fig. 5.11 y Fig. 5.12.

	150m	200m	250m	300m	325m	350m	375m	380m	385m	390m	392.5m
Antenas A	50	50	50	50	50	50	50	50	50	50	50
Antenas B	50	50	50	49.5	50	50	45	49	43.5	48	36.5
	395m	400m	550m	560m	570m	580m	590m	595m	600m	605m	610m
Antenas A	50	50	42.5	47	42	44.5	46	41.5	42	22.5	12
Antenas B	23	10	0	0	0	0	0	0	0	0	0

Tabla 5.6. Número de paquetes recibidos a diferentes distancias
(escenario 2)

Paquetes recibidos en escenario 2

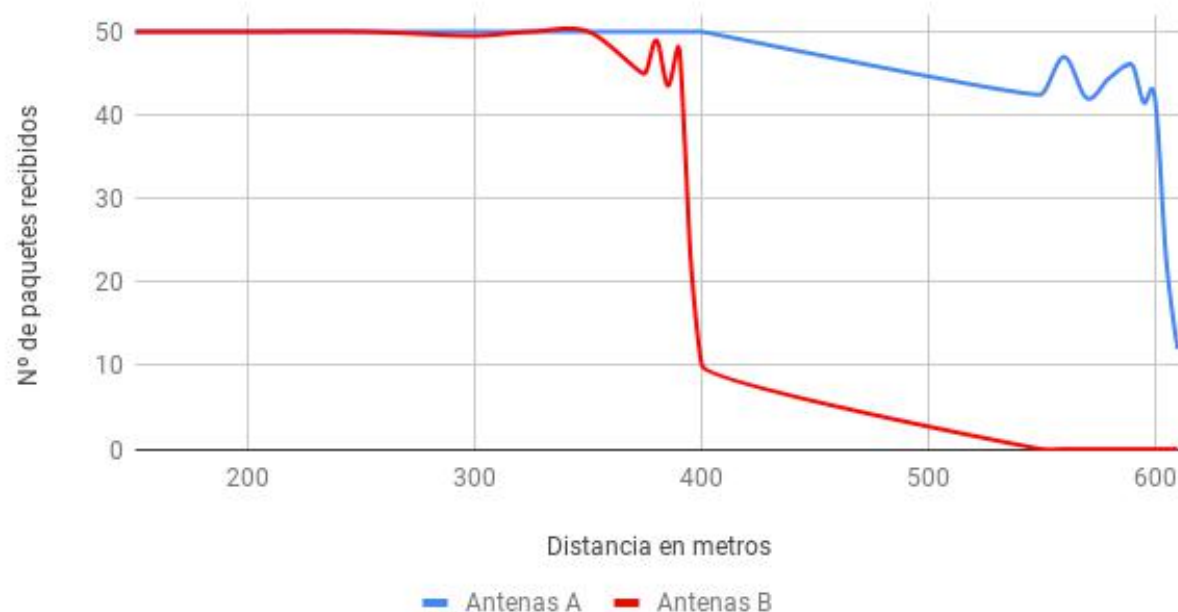


Fig. 5.11. Gráfica de número de paquetes recibidos a diferentes distancias (escenario 2)

	150m	200m	250m	300m	325m	350m	375m	380m	385m	390m	392.5m
Antenas A	0	0	0	0	0	0	0	0	0	0	0
Antenas B	0	0	0	0.01	0	0	0.1	0.02	0.13	0.04	0.27
	395m	400m	550m	560m	570m	580m	590m	595m	600m	605m	610m
Antenas A	0	0	0.15	0.06	0.16	0.11	0.08	0.17	0.16	0.55	0.76
Antenas B	0.54	0.8	1	1	1	1	1	1	1	1	1

Tabla 5.7. FLR en el escenario 2

Tras la realización de la prueba en este segundo escenario sin ningún posible obstáculo podemos sacar conclusiones claras. En primer lugar, la máxima distancia conseguida con las antenas B, recordemos que eran las más largas, ha sido de unos 395 metros aproximadamente, mientras que con las antenas A, las originales de los módulos nRF24L01, se pueden alcanzar los 600 metros. A partir de esas distancias el FLR supera el valor de 0.5, es decir, que se pierden más de la mitad de los paquetes transmitidos.

Para terminar, gracias a los datos recogidos podemos reafirmar que las antenas A funcionan mejor en nuestro sistema que las antenas B, como ya pudimos intuir en las pruebas realizadas en un laboratorio del departamento de TSC.

FLR (Escenario 2)

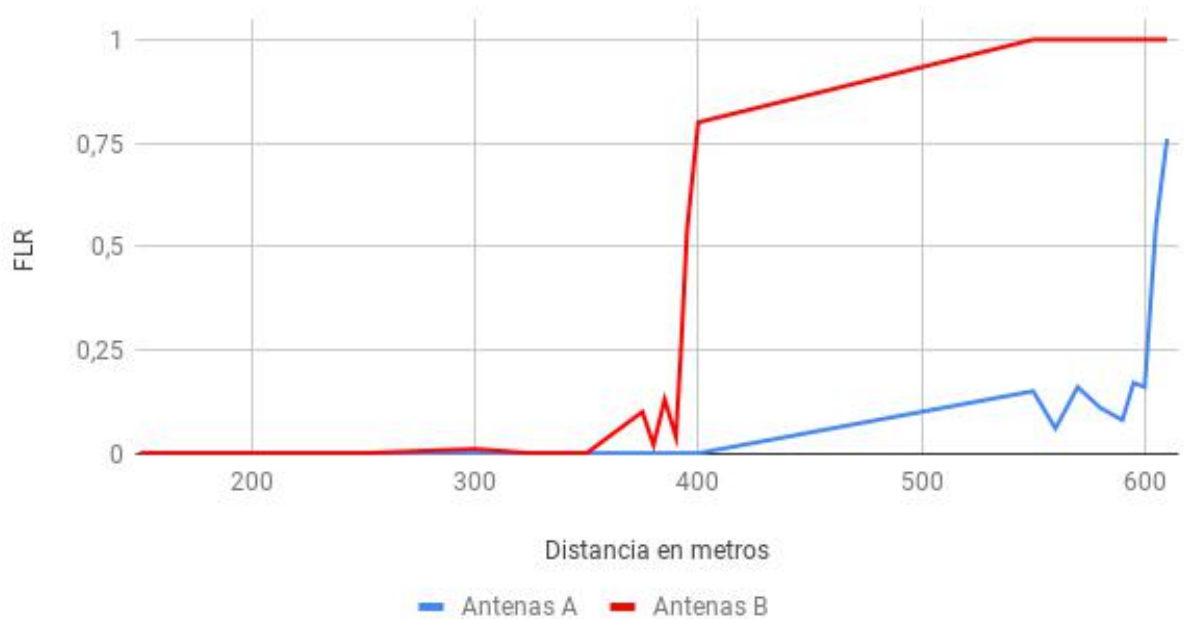


Fig. 5.12. Gráfica del FLR en el escenario 2

5.2. Resultados en la competición

Finalmente esta temporada sólo hemos podido asistir por problemas con las fechas de fabricación a la competición celebrada entre los días 21 y 25 de agosto en el circuito de Montmeló, denominada *Formula Student Spain*.

Durante estas fechas se han intentado hacer pruebas en la competición, pero cuando nos dimos cuenta de que el sistema no funcionaba correctamente, no hubo tiempo para desmontar y reparar el sistema debido a la programación oficial de las distintas pruebas de la competición.

Una vez de vuelta en la Universidad conseguimos detectar el problema y solucionarlo. A pesar de que actualmente el sistema funciona correctamente, no hemos podido tener la suerte de haber realizado pruebas en competición, pero se seguirá trabajando para que durante la próxima temporada sea posible.

En la Fig. 5.13 se muestra a todos los integrantes del equipo *Formula UC3M* que han acudido a la competición.



Fig. 5.13. Equipo *Formula UC3M* en la competición de *Formula Student Spain 2018*

6. PLANIFICACIÓN DEL TRABAJO

La realización de este proyecto comenzó el día 24 de Octubre de 2016 con la primera reunión con el tutor. A partir de ese mismo día se comenzó a pensar cómo construir el sistema de comunicaciones y a buscar posibles soluciones. A continuación se detallan las diferentes tareas que se han ido realizando, organizadas en seis bloques, con sus respectivos tiempos de ejecución.

Investigación (195 horas)

■ **Estudio previo (150 horas)**

Durante este período se propusieron posibles soluciones para el desarrollo del sistema y se analizaron las distintas posibilidades en cuanto a plataformas y tecnologías que se podrían utilizar.

■ **Contacto con empresas (45 horas)**

Entre las posibles soluciones se encontraba la de utilizar un sistema comercial que cumpliera con nuestras especificaciones. Por ello se contactó con casi 30 empresas diferentes.

Desarrollo del prototipo (57 horas)

■ **Primer ensamblaje (2 horas)**

El día que se contó con todos los elementos necesarios se comenzó a realizar el primer ensamblaje de los módulos nRF24L01 con las placas Arduino para probar su funcionamiento.

■ **Programación de un sistema emisor-receptor de prueba (10 horas)**

Tras terminar el primer ensamblaje se programaron programas sencillos para conocer, entender y probar el funcionamiento de los módulos nRF24L01.

■ **Programación de los distintos módulos del sistema (15 horas)**

Una vez que ya se conocía cómo trabajaban los módulos nRF24L01 y cómo interactuar con ellos, se comenzaron a programar nuestros programas principales de los módulos emisor y receptor.

■ **Realizar conexiones y ensamblaje del sistema completo (20 horas)**

Tras conseguir un prototipo funcional del sistema, era hora de preparar el ensamblaje definitivo del sistema, soldando las conexiones necesarias entre los cables y

los diferentes dispositivos y preparando los conectores de los módulos para facilitar su conexión una vez instalados en el coche.

- **Diseño e impresión en 3D de las cajas para los módulos (10 horas)**

Para evitar posibles daños en los distintos módulos del sistema de telemetría, se decidió diseñar pequeñas cajas que se pudieran imprimir en 3D para protegerlos.

Realización de pruebas (13 horas)

- **Realización de pruebas en laboratorio (3 horas)** Se realizaron las pruebas con el analizador de red, el generador de señales, la antena multibanda directica y el analizador de espectros en un laboratorio del departamento de TSC.

- **Realización de pruebas en el escenario 1 y su análisis (5 horas)**

Tras la realización de las pruebas de campo en el escenario 1 se llegó a la conclusión de que se debían repetir las pruebas en otro escenario más adecuado.

- **Realización de pruebas en el escenario 2 y su análisis (5 horas)**

Esta vez, las pruebas se realizaron satisfactoriamente y los resultados dieron la información suficiente para poder decir qué antenas daban mejor resultado.

Trabajo en el coche (85 horas)

- **Elección y colocación de sensores en el coche C3-R17 (60 horas)**

Se realizaron varias reuniones dentro del equipo para determinar qué sensores irían colocados finalmente en el coche. Una vez que ya se conocían los sensores que se querían instalar, se realizó la búsqueda de los mismos en el mercado.

Cuando se consiguieron todos los sensores se colocaron en el coche del año 2017, el C3-R17.

- **Instalación de sensores en el coche C3-R18 (15 horas)**

Una vez el prototipo de este año estuvo terminado, se procedió a desinstalar los sensores del C3-R17 para colocarlos en el modelo de este año, el C3-R18, junto con todos los cables necesarios para conectarlos al módulo principal.

- **Instalación del sistema de telemetría en el coche (10 horas)**

Tras tener los cuatro módulos del sistema terminados, se instalaron los módulos principal, datalogger y emisor en el coche y se realizaron pruebas de funcionamiento.

Competición (50 horas)

■ **Pruebas en competición (50 horas)**

Durante los días de la competición celebrada en Barcelona se realizaron pruebas del funcionamiento del sistema.

Memoria (300 horas)

● **Redacción de la memoria (300 horas)**

En la Fig. 6.1 se muestra el diagrama de Gantt que detalla todas las tareas anteriormente mencionadas. Incluye las fechas de inicio y de fin de cada tarea.

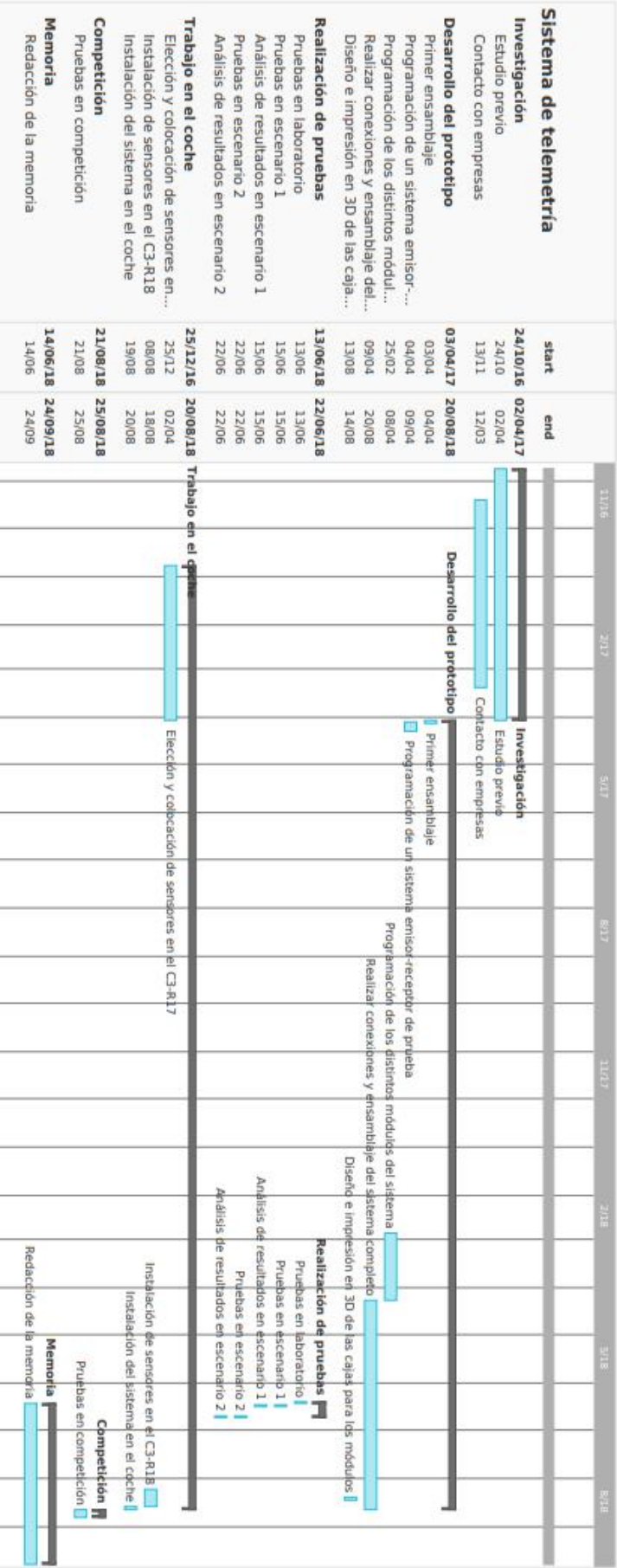


Fig. 6.1. Diagrama de Gantt del proyecto

7. PROBLEMAS ENCONTRADOS Y POSIBLES SOLUCIONES

7.1. Unidad de control de motor (ECU)

Inicialmente se había planteado como objetivo de este proyecto recopilar, además de los datos de los sensores, los datos recogidos por la unidad de motor o ECU. Finalmente no se llevó a cabo el desarrollo de esta última parte por la complejidad que añadiría al proyecto, el cual ya ha necesitado la inversión de mucho tiempo.

Para la próxima temporada se intentará incorporar la lectura de la información recogida por la ECU para tener más datos de diferentes partes del coche que ayuden al equipo a comprender y mejorar su rendimiento.

7.2. Sensores de efecto Hall

A pesar de tener planificada la recopilación de los datos de estos sensores, aunque se han colocado en el coche y están preparados para conectarse mediante un conector, finalmente no se han conectado a la placa Arduino Mega.

Esta decisión se tomó por el hecho de que los datos de estos sensores se utilizarán para realizar la configuración del coche, más concretamente del reparto de frenada, y para ello es necesario realizar pruebas con el coche en movimiento. Como no ha dado tiempo a rodar con el nuevo coche antes de la competición, no merecía la pena introducir más carga de procesamiento a la placa Arduino que pudiera ralentizar la ejecución del programa del módulo principal.

Por lo tanto, estos sensores se han dejado colocados y preparados para que en el futuro se puedan utilizar, pero no han sido conectados en la competición.

7.3. Situación de la antena en el monoplaça

Para la colocación de la antena se han tenido en cuenta muchos factores. Pero el más limitante ha sido la distancia entre la placa Arduino Mega y el módulo de radio emisor, ya que están comunicados mediante el protocolo SPI, el cual se ve afectado por la distancia de los cables, ya que es un protocolo de comunicación diseñado para comunicar elementos dentro de una misma placa, es decir, a corta distancia.

La primera opción que barajamos fue la de colocar la antena en el morro del coche, al igual que la llevan los coches de Fórmula Uno y otros muchos equipos de *Formula Student*. Esta primera opción era inviable por la distancia entre el módulo de radio y la placa Arduino Mega, debido al diseño de la comunicación SPI como

se ha comentado anteriormente. Además, queríamos evitar hacer un agujero en el monocasco siempre que fuera posible.

La decisión que tomamos fue la de probar la comunicación con una longitud considerable de los cables y comprobar si la comunicación entre módulos de radio funcionaba, e ir cortándolos poco a poco hasta conseguir que la comunicación fuera lo suficientemente fiable. De esta manera conseguimos mantener una distancia de unos 58-59 centímetros entre ambos módulos. Con esta distancia se intentó colocar en la parte más elevada y despejada de obstáculos posible del coche.

Una de las opciones era colocarlo en la parte delantera del coche, que está más despejada de piezas que puedas obstaculizar la comunicación, pero en los lugares a los que permitían los cables llegar podían molestar al piloto al entrar y salir del coche y, por tanto, probablemente se acabarían rompiendo por algún golpe.

Otra opción, que terminó siendo la definitiva, consistía en colocar el módulo de radio con la antena en uno de los brazos del arco principal o *main hoop* del sub-chasis tubular trasero, con la ayuda de un soporte especial impreso en 3D que se adaptara al tubo. De esta manera la antena se encuentra bastante cerca de tubos de acero por un lado, pero libre de obstáculos por el otro, y aunque no es la situación ideal, se decidió colocarla ahí de manera preliminar y realizar pruebas de su rendimiento. La situación final de la antena en el coche se muestra en la Fig. 7.1.

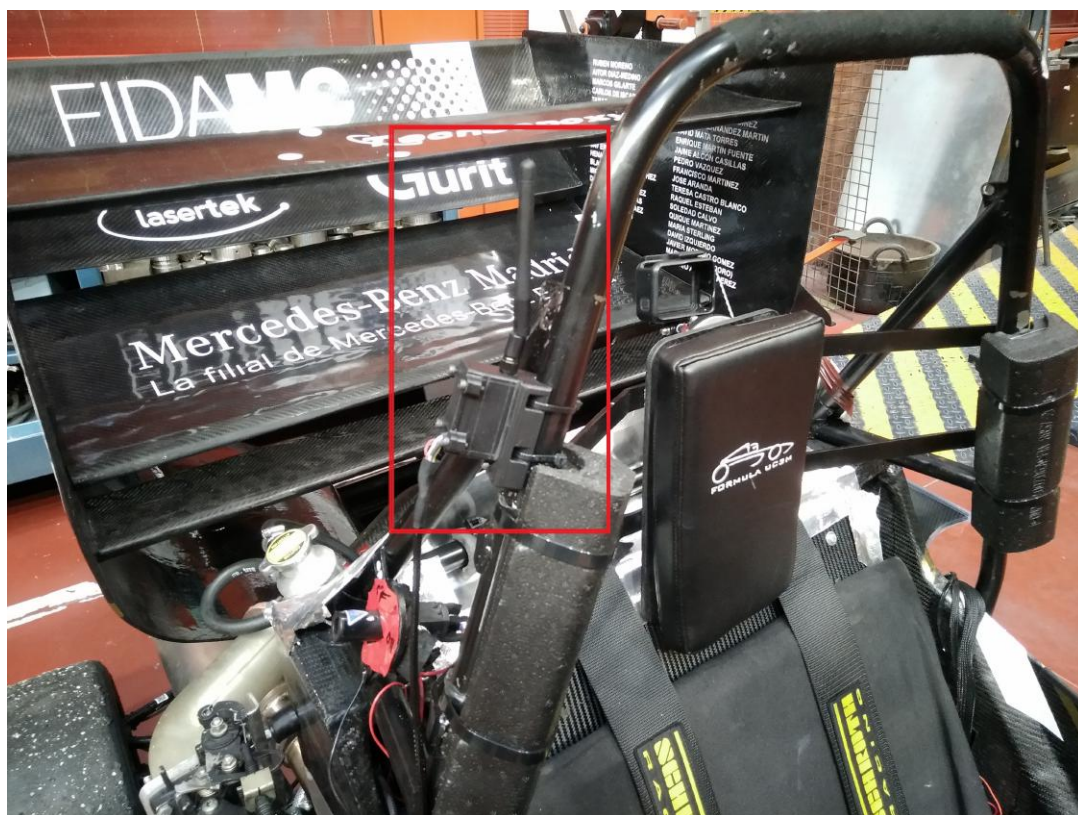


Fig. 7.1. Posición final de la antena en el monoplaça

7.4. Pruebas en competición

Pese a probar el correcto funcionamiento del sistema antes de viajar a Montmeló, como se ha comentado en apartados anteriores no se pudo probar su funcionamiento completo en la competición. Debido a problemas del hardware, no funcionó la transmisión mediante radio de los datos de los sensores recopilados.

Durante la competición se desmontó por completo el sistema de telemetría y se comprobaron todos los módulos por separado. Y a pesar de revisar todos los elementos y conexiones no logramos encontrar una solución.

Con la parte del datalogger también nos encontramos problemas. En este caso sí funcionó, ya que el programa se ejecutaba correctamente y se creaba un fichero de texto en la tarjeta micro-SD, pero a la hora de extraer la tarjeta e intentar leerla desde un ordenador no nos permitía abrir el fichero por error de formato. Para intentar corregirlo se formateó varias veces la tarjeta y se probó con distintas tarjetas y adaptadores de micro-SD a tarjeta SD, pero siempre se obtuvo el mismo resultado.

Anteriormente, el módulo de lector de tarjetas SD ya nos había dado ese problema alguna vez, pero formateando la tarjeta se solucionaba el problema. Esta vez sin embargo no pudimos arreglarlo durante la competición.

A la vuelta de la competición se probó a cambiar el *Shield* de Arduino para leer y escribir en tarjetas SD y se comprobó que los problemas obtenidos en la competición venían originados por el hardware. Probablemente provocados por la suciedad que había recibido durante las pruebas realizadas al inicio de esta temporada en el anterior coche, en las cuales se empezó a utilizar este módulo para recoger los datos de algunos sensores, y por las que se decidió fabricar una caja para proteger este módulo.

También se desmontaron y analizaron los módulos del emisor y el receptor a la vuelta de la competición para intentar averiguar el origen de los problemas. Sólo fallaba uno de los módulos, tanto configurado para operar de emisor como de receptor. El sistema volvió a funcionar cuando se reemplazó dicho módulo nRF24L01 por uno nuevo, por lo que seguramente fallara algún componente de la placa o alguna conexión de los pines que se soldaron durante el ensamblaje del sistema.

8. PRESUPUESTO

En este apartado vamos a cuantificar el coste económico del desarrollo de este proyecto. Para ello, analizaremos los costes tanto personales como los relacionados con los materiales y herramientas necesarias.

El cálculo de la amortización de los componentes utilizados se ha llevado a cabo mediante la siguiente fórmula:

$$\frac{t}{T} \times C$$

Donde:

t = tiempo de uso del componente

T= período de depreciación

C = coste del equipo

Personal	Tiempo (h)	Precio/hora (€/h)	Importe total (€)
Estudiante de ingeniería	700	30	21.000,00
Ingeniero	52	75	3.900,00
Técnico de laboratorio	5	30	150,00
TOTAL			25.050,00

Tabla 8.1. Costes de personal

Material	Tiempo usado (meses)	Período de depreciación (meses)	Importe (€)	Importe total (€)
PC más accesorios	20	48	750,00	312,50
Placa Arduino Mega	18	24	38,50	28,88
Placa Arduino Uno	18	24	22,00	16,50
Sensor Shield de Arduino	18	24	11,99	8,99
SD card Shield de Arduino	18	24	6,38	4,79
Módulos nRF24L01	18	24	8,50	6,37
Sensores amortiguadores KA Sensors KP13S	10	60	280,00	46,67
Sensores T ^a de Frenos MLX90614	10	36	22,00	6,11
Sensores de efecto Hall SP08 Wheel Speed Sensor	10	60	260,00	43,33
Tubo de Pitot PDA-8-F-6-KL	10	60	519,00	86,50
Termistores de 100k ohmios	10	12	1,50	1,25
Potenciómetros de 100k ohmios	10	12	4,00	3,33
Cables de automoción	6	12	5,00	2,50
Conectores Deutsch	6	12	85,00	42,50
Fundas termorretráctiles	6	12	3,00	1,50
Fundas de cable	6	12	10,00	5,00
Arduino IDE v.1.8.5	18	96	0,00	0,00
Telemetry Viewer v0.4	12	96	0,00	0,00
Plástico ABS	12	12	20,50	20,50
Impresora 3D Moebyus ONE	0.067	60	649,00	0,72
Antena multibanda directiva	0,033	180	15.000,00	2,75
Analizador de red HP Agilent 8753E	0,033	180	11.495,00	2,11
Generador de señales HP Agilent N5181A	0,033	180	7.925,00	1,45
Analizador de espectros HP 8594E	0,033	180	1.895,00	0,35
TOTAL				644,60

Tabla 8.2. Coste de material

Concepto	Importe total (€)
Coste de personal	25.050,00
Coste de materiales	644,60
Costes indirectos (20 %)	5.138,92
Subtotal	30.833,52
IVA (21 %)	6.475,04
TOTAL	37.308,56

Tabla 8.3. Costes totales

Por lo tanto, el coste total de este proyecto asciende a la cantidad de treinta y siete mil trescientos ocho euros con cincuenta y seis céntimos (37.308,56€).

9. CONCLUSIONES

Como valoración y análisis del resultado obtenido tras el desarrollo del sistema propuesto al inicio de este documento, y teniendo en cuenta tanto los aspectos positivos como los negativos, se han obtenido una serie de conclusiones que se desarrollarán a continuación.

9.1. Conclusiones generales

El objetivo de este proyecto ha sido diseñar y desarrollar un sistema de telemetría válido para un coche participante en la competición *Formula Student* y, aunque los resultados no han sido tan buenos como se esperaba, sí que ha cumplido con las principales características que se pedía, las cuales consistían en conseguir un sistema autónomo que no dependiera de su localización para su funcionamiento, que fuera un sistema sencillo, económico y que se pueda reutilizar en futuras temporadas.

A falta de probar el sistema con el coche en movimiento, no podemos estar seguros de que nuestro proyecto cumple todas las expectativas de rendimiento deseadas. Por ello, sean cuales sean los estos resultados, tenemos la posibilidad de mejorar los inconvenientes que se han encontrado como trabajo para el futuro.

9.2. Futuras líneas de trabajo

El principal objetivo a partir de ahora es conseguir hacer que el sistema funcione colocado en el coche y con todos los sensores conectados. Aunque se ha conseguido hacer que funcione de manera independiente al coche, queda probarlo con las mangueras de cables y los sensores instalados en él.

Para próximas temporadas la intención es aumentar el número de sensores en el coche, como por ejemplo conectar los sensores de efecto Hall o colocar un potenciómetro para calcular la posición del volante para analizar y comparar el tipo de conducción de los distintos pilotos. Pero antes de eso hay que conseguir que su rendimiento con el coche en movimiento sea adecuado para empezar a mejorar el sistema partiendo de la base de que es un sistema válido para nosotros.

Además, como se ha comentado en el apartado anterior, se debe estudiar cómo suplir las limitaciones encontradas en el sistema. Por ejemplo, con los módulos nRF24L01 no se pueden mandar tramas con más de 32 bytes de información. Con el número de sensores que tenemos actualmente instalados nos es suficiente, pero si en el futuro se quiere incrementar el número de sensores en el coche como se

ha comentado anteriormente, habría que utilizar dos módulos independientes que transmitieran en paralelo o buscar otra tecnología que no tenga esa limitación.

Y por último, nos gustaría poder desarrollar una interfaz gráfica propia y personalizada para el equipo, como se han observado en proyectos similares, analizados en el apartado 2.3.

BIBLIOGRAFÍA

- [1] Internet of Things Related Standards, «Internet of Things» [En línea]. Disponible en: <https://standards.ieee.org/initiatives/iot/stds.html>. [Último acceso: 19 Septiembre 2018].
- [2] 3GPP Specification series, «5G» [En línea]. Disponible en: <http://www.3gpp.org/DynaReport/38-series.htm>. [Último acceso: 19 Septiembre 2018].
- [3] Tablas de atribución de frecuencias, «CNAF» [En línea]. Disponible en: http://www.mincotur.gob.es/telecomunicaciones/espectro/CNAF/tablas_2017.pdf. [Último acceso: 22 Septiembre 2018].
- [4] Boletín Oficial del Estado (BOE) del día 09/05/2013, actualizado el día 27/10/2017. [En línea]. Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2013-4845>. [Último acceso: 21 Septiembre 2018].
- [5] PEDRO IGLESIAS ORTIZ, "Telemetría de vehículos FSAE con PIC32 y transceptor de radio CC110L", Proyecto fin de carrera, Dpto. de Sistemas Electrónicos y de Control, Universidad Politécnica de Madrid, Madrid, España, 2015. [En línea]. Disponible en: http://oa.upm.es/40126/3/PFC_PEDRO_IGLESIAS_ORTIZ_1.pdf. [Último acceso: 20 Septiembre 2018].
- [6] JAIME IGLESIAS HERNÁNDEZ, "Sistema de telemetría para un vehículo de Fórmula S.A.E.", Trabajo fin de grado, Dpto. de Sistemas Electrónicos y de Control, Universidad Politécnica de Madrid, Madrid, España, 2013. [En línea]. Disponible en: http://oa.upm.es/21970/1/TFG_JAIME_IGLESIAS_HERNANDEZ.pdf. [Último acceso: 20 Septiembre 2018].
- [7] PEDRO C. LÓPEZ JIMÉNEZ, "Desarrollo de un dispositivo de telemetría y geolocalización basado en la plataforma Arduino y Shield 3G+GPS", Trabajo fin de grado, Dpto. de Electrónica, Tecnología de Computadoras y Proyectos, Universidad Politécnica de Cartagena, Murcia, España, 2014. [En línea]. Disponible en: <http://repositorio.upct.es/bitstream/handle/10317/4593/tfg338.pdf;sequence=1>. [Último acceso: 20 Septiembre 2018].
- [8] DANIEL GÓMEZ SANZ, "Diseño de un sistema de telemetría para un monoplaza de Formula Student", Proyecto fin de carrera, Dpto. de Ingeniería Electrónica, Universidad de Sevilla, Andalucía, España, 2015. [En línea]. Disponible en: <http://bibing.us.es/proyectos/abreproy/12303/fichero/pfcTipoETSI.pdf>. [Último acceso: 20 Septiembre 2018].

- [9] IEEE 802.15 WPAN Task Group 4 (TG4), «ZigBee» [En línea]. Disponible en: <http://www.ieee802.org/15/pub/TG4.html>. [Último acceso: 19 Septiembre 2018].
- [10] SigFox, «SigFox Technical Overview» [En línea]. Disponible en: <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>. [Último acceso: 19 Septiembre 2018].
- [11] Nordic Semiconductor, «nRF24L01. Single Chip 2.4GHz Transceiver. Preliminary Production Specification v1.0» [En línea]. Disponible en: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf. [Último acceso: 22 Septiembre 2018].
- [12] WEI-YEN LIN, MEI-WEN LI, KUN-CHAN LAN, CHUNG-HSIEN HSU, "A comparison of 802.11a and 802.11p for V-to-I communication: a measurement study", Paper de investigación, Dpto. de Computer Science and Information Engineering, Universidad de Vanung, Tainan City, Taiwan, 2010. [En línea]. Disponible en: [http://www.csie.ncku.edu.tw/klan/data/paper/pdf/A %20comparison %20of %20802.11a %20and %20802.11p %20for %20V-to-I %20communication_ %20a %20measurement %20study.pdf](http://www.csie.ncku.edu.tw/klan/data/paper/pdf/A%20comparison%20of%20802.11a%20and%20802.11p%20for%20V-to-I%20communication_%20a%20measurement%20study.pdf). [Último acceso: 20 Septiembre 2018].
- [13] Página oficial de Arduino «ARDUINO MEGA 2560 REV3» [En línea]. Disponible en: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>. [Último acceso: 21 Septiembre 2018].
- [14] FARREL FARAHBOD, "Telemetry Viewer v0.4", *Farrell's Blog*, 21-07-2017. [En línea]. Disponible en: http://www.farrellf.com/projects/software/2017-07-21_Telemetry_Viewer_v0.4/. [Último acceso: 25 Septiembre 2018].
También disponible en: <https://github.com/farrellf/TelemetryViewer>. [Último acceso: 25 Septiembre 2018].

ANEXO A: PRESUPUESTOS SOLUCIONES COMERCIALES

PRESUPUESTO 1



Quotation

Av. de la Universidad
Av. de la Universidad, 30
Leganes, , Madrid 28911
Spain

Quote Date
28/02/2017

Quote Number
CWQ11180

Cohda Wireless Europe GmbH
VAT ID DE292537652

Lead Time: 2 Weeks

Description	Quantity	Unit Price	Discount	AmountEURO
MK5 OBU KIT INCLUDING 1 YEAR MAINTENANCE	1	1,880.00	1,598.00	282.00
SOFTWARE SDK Release 13	1	3,240.00	2,430.00	810.00
Shipping & Handling	1	123.00	0.00	123.00
Subtotal				1,215.00
Total Discount				4,028.00
VAT Free Sales				0.00
Quote Total EURO				1,215.00

Special Price to support the University Project

Payment Terms: 30 days

PRESUPUESTO 2

Raveon Technologies Corporation
2320 Cousteau Court
Vista, CA 92081 USA
Ph:1-760-444-5995 Fax:1-760-444-5997
Email: sales@raveon.com



Quote

Customer **Carlos III University of Madrid**
Name Ruben Moreno
Address Calle Madrid 126
28903 Getafe Madrid Spain
rubenmoreno_94@hotmail.com

Quote **C3UM001**
Date 12/2/2016
EXW Vista, CA

Item	Qty	Part Number	Description	Available /LT ARO	Unit Price	TOTAL
Equipment for Vehicle						
	1	RV-M22S-ED	Tech Series license free wireless modem, with 3m power cord, 7-30V DC input, RS232, DB9 IO, D50 modem, 868 MHz (Europe),	4-6 Weeks	\$ 194.00	\$ 194.00
	1	RA-A96-EC	Antenna, Whip, 824 - 960 MHz, Mag Mount, 2 dBi, SMA	4-6 Weeks	\$ 11.00	\$ 11.00
	1	RK-E06	Adapter, Interseries, TNC male to SMA female	4-6 Weeks	\$ 5.00	\$ 5.00
	1	RT-AN-GP1	Antenna, GPS, Magnet mount, SMA conn, 6', 3.3V.	4-6 Weeks	\$ 30.00	\$ 30.00
RV-M22 GPS Transceiver to be provided with 7 to 30 Vdc (3 W) using power cord included. RS-232 I/O Interface quoted. Other options available upon request. Antennas for the vehicle are specified as mag mount and include 3 m cables. Other antenna options available upon request. Installation is not included.						
Equipment for Pit Area Base Station						
	1	RV-M22S-ED	Tech Series license free wireless modem, with 3m power cord, 7-30V DC input, RS232, DB9 IO, D50 modem, 868 MHz (Europe),	4-6 Weeks	\$ 194.00	\$ 194.00
	1	PS-60-M21	Power Supply, AC, 12V, 5A, UL&CE, with connector for M21 or M22 modem	4-6 Weeks	\$ 74.00	\$ 74.00
	1	RA-A97-EC	Antenna, Base, Omni, 824 - 960 MHz, 3.5 dBi, N-female	4-6 Weeks	\$ 149.00	\$ 149.00
	1	RT-AN-GP1	Antenna, GPS, Magnet mount, SMA conn, 6', 3.3V.	4-6 Weeks	\$ 30.00	\$ 30.00
	1	RT-AN-395	500 Watt Flange Mt Lightning Arrestor w/N-Females	4-6 Weeks	\$ 119.00	\$ 119.00
RV-M22 GPS Transceiver to be provided with 7 to 30 Vdc (3 W). Power supply is quoted. RS-232 I/O Interface is quoted. Other options available upon request. RF Antenna should be mounted as high as possible on a mast or pole. For permanent installations, a properly grounded RT-AN-395 lightning arrestor is recommended where RF Cable enters building or enclosure. RF Cable and Installation is not included						

SubTotal \$ 806.00

Est Shipping and Handling

Tax Rate(s)

TOTAL \$ 806.00

Comments

Other

Ex Works Vista.
Freight not estimated. You may specify a freight account.
Prices quoted are in US Dollars (US\$)
Delivery per mutually agreed schedule
to be determined at time of order.

*Prices valid for 30 days from date of issue.

TERMS: 100% at time of order. Credit Card orders accepted via phone, with email confirmation. All prices quoted are in U.S. Dollars.

Unless otherwise agreed to in writing and signed by an officer of Raveon Technologies, any order, if accepted, is accepted subject to these Raveon Standard Terms and Conditions of Sale ("T&Cs"). They are available for review at: <http://www.raveon.com/terms.html>

Wire Transfer Instructions:

Chase Bank
Account Name: Raveon Technologies Corporation
Account Number: 213963372
Contact: John Wiskosky
2219 Palomar Airport Road
Carlsbad, California 92011
Routing transit number: 322271627
SWIFT address: CHASUS33
Phone: 1-760-431-8130 Fax: 1-877-501-1408
Customer responsible for fees of originating bank

Curt Buck x121 cb@raveon.com

Raveon Technologies Corporation - Advancing the Art of Wireless

ANEXO B: CÓDIGO ARDUINO

CÓDIGO DEL MÓDULO EMISOR:

```
1  #include <EEPROM.h>
2  #include <SD.h>
3  #include <SPI.h>
4  #include <i2cmaster.h>
5  #include <nRF24L01.h>
6  #include <RF24.h>
7
8  //RADIO
9  RF24 radio(9, 10); // CE, CSN pins
10 const byte channel[6] = "00001"; //channel of communication
11 const uint8_t array_size = 13;
12
13 struct data_struct
14 {
15     long time;
16     short sens[array_size];
17 }
18 data;
19
20 //SD
21 const int chipSelect = 49;
22 File logFile;
23
24 //AMORTIGUADORES
25 const int shockAbsFrontRight = A0;
26 const int shockAbsFrontLeft = A1;
27 const int shockAbsRearRight = A2;
28 const int shockAbsRearLeft = A3;
29
30 const int shockAbsFrontRight_offset = 95;
31 const int shockAbsFrontLeft_offset = 63;
32 const int shockAbsRearRight_offset = 220;
33 const int shockAbsRearLeft_offset = 194;
34 //VALOR AMORTIGUADORES
35 int shockAbsFrontRight_val = 0;
36 int shockAbsFrontLeft_val = 0;
37 int shockAbsRearRight_val = 0;
38 int shockAbsRearLeft_val = 0;
39
40 //TERMISTORES RADIADORES
41 const uint8_t radiatorTemp1 = A4;
42 const uint8_t radiatorTemp2 = A5;
43 const uint8_t radiatorTemp3 = A6;
```

```

44 //TEMPERATURA RADIADORES
45 uint8_t radiatorTemp1_val = 0;
46 uint8_t radiatorTemp2_val = 0;
47 uint8_t radiatorTemp3_val = 0;
48
49 //TERMISTORES RODAMIENTOS MANGUETAS
50 const uint8_t uprightRearRight = A8;
51 const uint8_t uprightRearLeft= A9;
52 //TEMPERATURA RODAMIENTOS MANGUETAS
53 uint8_t uprightRearRight_val = 0;
54 uint8_t uprightRearLeft_val = 0;
55
56 //FRENOS
57 const uint8_t brakeFrontRight = 0x34<<1; //i2c directions
58 const uint8_t brakeFrontLeft = 0x50<<1;
59 const uint8_t brakeRearRight = 0x4C<<1;
60 const uint8_t brakeRearLeft = 0x20<<1;
61 //TEMPERATURA FRENOS
62 uint8_t brakeFrontRight_val = 0;
63 uint8_t brakeFrontLeft_val = 0;
64 uint8_t brakeRearRight_val = 0;
65 uint8_t brakeRearLeft_val = 0;
66
67 //PEDALES ACELERADOR Y FRENO
68 const int throttlePedalPosition = A12;
69 const int brakePedalPosition = A13;
70 // % PEDALES ACELERADOR Y FRENO
71 uint8_t throttlePedal_percent = 0;
72 uint8_t brakePedal_percent = 0;
73
74 //PITOT
75 const int pitotSignal=A15;
76 float pitot=0;
77 float pitotTension;
78 float airPression;
79 float velocity;
80 const int P=101325; // Ideal Atmospheric pressure
81 const int pAtm=101325; // Atmospheric pressure
82 const int temp_amb=298; // Ambient temperature in kelvin at 25 C
83
84 void setup()
85 {
86     Serial.begin(9600);
87     Serial.print("Initializing SD card...");
88
89     if (!SD.begin(chipSelect))
90     {
91         Serial.println("Card failed, or not present");
92         return;
93     }
94     logFile = SD.open("DATALOG.txt", FILE_WRITE);

```

```

95     Serial.println("card initialized.");
96
97     pinMode (shockAbsFrontRight, INPUT);
98     pinMode (shockAbsFrontLeft, INPUT);
99     pinMode (shockAbsRearRight, INPUT);
100    pinMode (shockAbsRearLeft, INPUT);
101
102    i2c_init(); // Starts the i2c
103    bus. // Enable pull-ups.
104
105    radio.begin();
106    radio.openWritingPipe(channel);
107    radio.setPALevel(RF24_PA_MIN);
108    radio.stopListening();
109 }
110
111 void loop()
112 {
113
114     //AMORTIGUADORES
115     shockAbsFrontRight_val=analogRead (shockAbsFrontRight)-
116     shockAbsFrontRight_offset;
117     shockAbsFrontLeft_val=analogRead (shockAbsFrontLeft)-
118     shockAbsFrontLeft_offset;
119     shockAbsRearRight_val=analogRead (shockAbsRearRight)-
120     shockAbsRearRight_offset;
121     shockAbsRearLeft_val=analogRead (shockAbsRearLeft)-
122     shockAbsRearLeft_offset;
123     //TERMOPARES RADIADORES
124     radiatorTemp1_val=analogRead(radiatorTemp1);
125     radiatorTemp2_val=analogRead(radiatorTemp2);
126     radiatorTemp3_val=analogRead(radiatorTemp3);
127     //PEDALES
128     throttlePedal_percent=pedalsPercentage(throttlePedalPosition);
129     brakePedal_percent=pedalsPercentage(brakePedalPosition);
130
131     //PITOT
132     pitot=analogRead(pitotSignal);
133
134     pitotTension=map(pitot, 0, 1023, 0, 500);
135     airPression=((pitotTension/100)/(5-0.5))/(0.2*1000));
136     velocity=sqrt(574.1*airPression*(temp_amb/pAtm));
137
138     // TEMPERATURA FRENOS
139     brakeFrontRight_val = temperatureCelcius(brakeFrontRight);
140     brakeFrontLeft_val = temperatureCelcius(brakeFrontLeft);
141     brakeRearRight_val = temperatureCelcius(brakeRearRight);
142     brakeRearLeft_val = temperatureCelcius(brakeRearLeft);
143
144     //BUILD THE DATA FRAME

```

```

141     data.time= millis();
142
143     //TEMP FRENOS
144     data.sens[0] = brakeFrontRight_val;
145     data.sens[1] = brakeFrontLeft_val;
146     data.sens[2] = brakeRearRight_val;
147     data.sens[3] = brakeRearLeft_val;
148
149     //TEMP RADIADORES
150     data.sens[4] = radiatorTemp1_val;
151     data.sens[5] = radiatorTemp2_val;
152     data.sens[6] = radiatorTemp3_val;
153
154     //TEMP MANGUETAS
155     data.sens[7] = uprightRearRight_val;
156     data.sens[8] = uprightRearLeft_val;
157
158     //POSICION PEDALES
159     data.sens[9] = throttlePedal_percent;
160     data.sens[10] = brakePedal_percent;
161
162     //PITOT
163     data.sens[11] = pitotTension;
164     data.sens[12] = airPression;
165     data.sens[13] = velocity;
166
167     if (radio.write(&data, sizeof data))
168     {
169         Serial.println("Data sent!");
170     }
171
172     //SD
173
174     if (logFile)
175     {
176         //REVISAR lineas /columnas de escritura
177         logFile.print(millis());
178         logFile.print("\t");
179         logFile.print(shockAbsFrontRight_val); //AMORTIGUADORES
180         logFile.print("\t");
181         logFile.print(shockAbsFrontLeft_val);
182         logFile.print("\t");
183         logFile.print(shockAbsRearRight_val);
184         logFile.print("\t");
185         logFile.print(shockAbsRearLeft_val);
186         logFile.print("\t");
187         logFile.print(brakeFrontRight_val); //FRENOS
188         logFile.print("\t");
189         logFile.print(brakeFrontLeft_val);
190         logFile.print("\t");
191         logFile.print(brakeRearRight_val);

```

```

192     logFile.print("\t");
193     logFile.print(brakeRearLeft_val);
194     logFile.print("\t");
195     logFile.print(radiatorTemp1_val); //TERMOPARES
196     logFile.print("\t");
197     logFile.print(radiatorTemp2_val);
198     logFile.print("\t");
199     logFile.print(radiatorTemp3_val);
200     logFile.print("\t");
201     logFile.print(uprightRearRight_val);
202     logFile.print("\t");
203     logFile.print(uprightRearLeft_val);
204     logFile.print("\t");
205     logFile.print(throttlePedal_percent); //PEDALES
206     logFile.print("\t");
207     logFile.print(brakePedal_percent);
208     logFile.print("\t");
209     logFile.print(pitotTension); //PITOT
210     logFile.print("\t");
211     logFile.print(airPression);
212     logFile.print("\t");
213     logFile.print(velocity);
214     logFile.println("\t");
215     logFile.close();
216     Serial.println("Saved in SD card");
217
218 }
219 else
220 {
221     Serial.println("Error opening file");
222 }
223 }
224
225 /**
226  * Converts the read of the breaks temperature sensors to celsius
227  * grades
228  * @param addres Addres of the sensor
229  * @return celsius The value of the sensor's read in celcius
230  * degrees
231  */
232 float temperatureCelcius(int address)
233 {
234     int dev = address;
235     int data_low = 0;
236     int data_high = 0;
237     int pec = 0;
238
239     i2c_start_wait(dev+I2C_WRITE);
240     i2c_write(0x07);
241
242     i2c_rep_start(dev+I2C_READ);

```



```

241     data_low = i2c_readAck();
242     data_high = i2c_readAck();
243     pec = i2c_readNak();
244     i2c_stop();
245
246     double tempFactor = 0.02;           // resolution of MLX90614
247     double tempData = 0x0000;
248
249     tempData = (double)(((data_high & 0x007F) << 8) + data_low);
250     tempData = (tempData * tempFactor)-0.01;
251     float celcius = tempData - 273.15; //kelvin to celsius
252     return celcius;
253 }
254
255 /**
256  * Converts the value read by potentiometer to percentage
257  * @param value Value of the potentiometer
258  * @return percentage The equivalent percentage to the
259  *         potentiometer value
260  */
261 uint8_t pedalsPercentage(int value)
262 {
263     uint8_t percentage = (value/1024.00)*100;
264     return percentage;
265 }

```

CÓDIGO DEL MÓDULO RECEPTOR:

```
1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4
5  //RADIO
6  RF24 radio(7, 8); // CE, CSN pins
7  const byte channel[6] = "00001"; //channel of communication
8
9  const uint8_t array_size = 13;
10
11 struct data_struct
12 {
13     long time;
14     short sens[array_size];
15 }
16 data;
17
18 void setup()
19 {
20     Serial.begin(9600);
21     radio.begin();
22     radio.openReadingPipe(0, channel);
23     radio.setPALevel(RF24_PA_MIN);
24     radio.startListening();
25 }
26 void loop()
27 {
28
29     if (radio.available())
30     {
31         radio.read(&data, sizeof (data));
32         Serial.print(data.time);
33
34         for(int i=0; i<array_size; i++)
35         {
36             Serial.print(",");
37             Serial.print(data.sens[i]);
38         }
39         Serial.println();
40     }
41 }
```

ANEXO C: CÓDIGO ARDUINO PRUEBAS

CÓDIGO DEL MÓDULO EMISOR:

```
1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4
5  //RADIO
6  RF24 radio(9, 10); // pines CE, CSN
7  const byte address1[6] = "00001";
8  const uint8_t tam_array = 32;
9
10 struct datos
11 {
12     //float time;
13     uint8_t sens[tam_array];
14 }
15 data;
16
17 int contador = 1;
18
19 void setup()
20 {
21
22     Serial.begin(9600);
23     radio.begin();
24     radio.stopListening();
25     radio.openWritingPipe(address1);
26     radio.setPALevel(RF24_PA_MAX);
27     //radio.setCRCLength( RF24_CRC_8 ); // 8 bits CRC
28     radio.setPayloadSize(32); //8, 16 o 32 bytes
29     Serial.println("Initializing SD card...");
30     i2c_init(); // Inicia el bus i2c
31     .
32     PORTC = (1 << PORTC4) | (1 << PORTC5); // Habilita
33     pullups .
34 }
35
36 void loop()
37 {
38     //while(1)
39     {
40         while(contador<=50)
41         {
42             //RADIO
43             //data.time= millis();
```

```
42
43     for(short i=0; i<tam_array; i++)
44     {
45         data.sens[i] = i+1;
46     }
47
48     Serial.println(radio.isChipConnected());
49     Serial.println(radio.write(&data, sizeof data));
50
51     if(radio.write(&data, sizeof data))
52     {
53         Serial.print("Enviado ");
54         Serial.println(contador);
55     }
56     delay(245);
57
58     contador++;
59 }
60
61 //Fin del loop
```

CÓDIGO DEL MÓDULO RECEPTOR:

```
1  #include <SPI.h>
2  #include <nRF24L01.h>
3  #include <RF24.h>
4  RF24 radio(9, 10); // CE, CSN
5  const byte address1[6] = "00001";
6
7  const uint8_t tam_array = 32;
8
9  struct datos
10 {
11     //float time;
12     uint8_t sens[tam_array];
13 }
14 data;
15
16 void setup()
17 {
18     Serial.begin(9600);
19     radio.begin();
20     radio.openReadingPipe(0, address1);
21     radio.setPALevel(RF24_PA_MAX);
22     //radio.setCRCLength( RF24_CRC_8 ); // 8 bits CRC
23     radio.startListening();
24 }
25 void loop()
26 {
27     Serial.println(radio.isChipConnected());
28     if (radio.available())
29     {
30         Serial.println("RADIO OK");
31         radio.read(&data, sizeof data);
32         //Serial.print(data.time);
33
34         for(int i=0; i<tam_array; i++)
35         {
36             Serial.print(data.sens[i]);
37             Serial.print("\t");
38         }
39         Serial.println();
40     }
41     //delay(500);
42 }
```

ANEXO D: ENGLISH SUMMARY

Introduction

The main objective of this project is to develop a telemetry system for Formula UC3M team, which is a group of students that design and build a race car to compete in the interuniversity world competition Formula Student. This year prototype is shown in Fig. 9.1.

Formula UC3M is the team that represents our University in this competition. It is formed by students from different degrees of our University, be it bachelor's degrees like Mechanical Engineering, Economics or masters and PhD.

The telemetry system will be able to read real time data from sensors located in the car, record them and send them to a receiver device. The car and the receiver device could be distanced at about 400 metres.

To develop the project we will try to use a radio communication link. For that we will use frequencies that belong to ISM band, which are frequencies that do not need extra permission from authorities to be used. ISM means Industrial, Scientific and Medical, and were created to be free use frequencies. For example, Wi-Fi and Bluetooth technologies work in one of these band of frequencies.



Fig. 9.1. C3-R18 prototype car

System design

First of all we need to know which sensors will we use to control the car behavior while it is racing. Next, all the sensors will be listed:

- **Shock absorbers (x4)** (*Model: KA Sensors KP13S*)

The data from this sensors will help us to know how the car behaves. With this information we could setup the suspension system later.

- **Brakes temperature (x4)** (*Model: MLX90614*)

This information is to avoid the overheating of the brakes during the competition.

- **Rear Uprights temperature (x2)** (*Model: comun 100k ohms thermistor*)

In this case the sensor data is also to avoid possible breaks due to overheating of the bearings between the uprights and the wheel hubs. This two parts connect the wheel set to the suspension set.

- **Refrigeration system temperature (x3)** (*Model: comun 100k ohms thermistor*)

For us is important to know if the refrigeration system is working correctly. For this reason we will put three temperature sensors or thermocouples: one in the beginning of the refrigeration system, the second between the two radiators that we have installed in the car, and the last one at the exit of the second radiator.

- **Pedals position (x2)** (*Model: comun 100k ohms potentiometer*)

With two potentiometers we will know in which position are the pedals. This information will help us to compare the driving style between the different drivers.

- **Wheels speed (x4)** (*Model: SP08 Wheel Speed Sensor*)

The data from these sensors its needed to adjust the brake distribution.

- **Pitot tube (x1)** (*Model: PDA-8-F-6-KL*)

Pitot tube is used to know the air pression and the velocity of the air in a determined point. We will use it to test the aerodynamics parts of our car.

Now that we know the data that we have to send, we need to choose which technology and devices will we use to develop this project. To do a decision we are going to compare between different possibilities. These are the different devices that we are going to consider to develop our system:

- **Commercial solutions**

The first option is to search some commercial solution that could fix and satisfy our needs. We contacted with 29 companies, but no one was a good

option for us, because of their prices or because their products did not fit with what we were looking for, or because they did not answer us simply.

Some of these budgets are attached in annex A.

- **USRP**

Universal Software Radio Peripheral, or USRP, are radio devices that can be configured by LabVIEW software. Its main disadvantage is its high economic cost. For this reason, we discarded this option.

- **Raspberry Pi**

Raspberry Pi is an open hardware platform that offers a lot of versatility. Because of that and for its prices it could be a good decision to choose this option to implement our telemetry system with.

- **Arduino**

As well as Raspberry Pi, Arduino Pi is an open hardware platform that offers many complementary modules to add to the main board, giving it possibilities to front facing many kind of projects.

Between all the previous options we have decided to use Arduino boards to develop our system because of its prices and because we used Arduino's boards previous years, so we have experience programming and testing them.

Finally, we need to choose a technology to send our data. The technologies we can use with Arduino are:

- **Bluetooth**

Bluetooth works in the ISM band of 2.4 GHz.

- **Wi-Fi**

- **WiMax**

- **Mobile network**

This technology offers a big range of coverage, but it depends of the mobile network's coverage and it is a disadvantage. The editions of Formula Student are celebrated in automotive circuits, where the mobile network coverages are usually poor, and we need an autonomous system that does not depend on external factors. For this reason, we discard this option.

- **ZigBee**

- **XBee transceivers**

- **SigFox**

- **Other solutions working on ISM bands**

In this category we find nRF24L01 transceiver, which use an own protocol and reaches a coverage of 600 metres approximately. It is compatible with

Arduino and very cheap. For this reasons we have choosen this option as the technology that we will use.

System development

To develop the project we will use two Arduino boards, Arduino Uno and Arduino Mega, and two Arduino extra modules to complement them: a SD card reader module and the Sensor Shield for Arduino Mega board, to connect all the sensors to it easily.

The system will be formed for 4 different modules:

The main module, formed by the Arduino Mega board and its Sensor Shield, will be located in the car, beside the driver's seat. Together with it will be the datalogger module, composed by the SD card module, connected to the main module.

The other two modules are the transmitter and receiver modules. The first one will be also connected to the main module and it is formed by a nRF24L01 transceiver and its antenna. The last module, the receiver, its composed by an Arduino Uno board and another nRF24L01 transceiver, that will receive the data sent from the nRF24L01 module of the transmitter module.

To develop the whole system we are going to use Arduino's IDE to programm the boards and a programm called Telemetry Viewer [14] to visualize the data sent from the transmitter module.

System tests

To test the range of coverage that our nRF24L01 modules could achieve we performed tests in open areas.

Firstly, we tried them in a wide area in the town of Getafe, but we only was able to distance the transmitter and the receiver 400 metres due to physical limitations. Furthermore, in the last stretch of the test, we could not avoid that some trees would be between the transmitter and receiver line of vision. Because of that, we found some irregularities in the results.

We repeated the test in a bigger area that guaranteed a direct line of vision between the two transceivers modules. In this case, we were able to finish the tests successfully, achieving a range of 605 metres in which the receiver was able to receive more than half messages that was sent.

Found problems

After finish the prototype we tested it before install it in the car, but when we were in Barcelona, at the 2018 Formula Student Spain Edition, we had problems with the

telemetry system.

We could not do it works. It did not send or receive any sensor data. We disassembled the whole system and tried every single module, but we could not find the origin of the problem. The datalogger module worked, but we could not read the file from SD card because of a failure with the file format. We tried formatting the SD card but the problem persisted.

When we came back from the competition, we could find out the problem. In both cases, it was a hardware problem. We replaced the SD card module with a new one and it worked properly. We did the same with the transmitter module, which was the module that did not work in the competition, but with the new nRF24L01 transceiver did it.

General conclusions

The objective was to design and develop a telemetry system for the Formula UC3M team's race car, and as global valuation, the objective of the project was fulfilled, in absence of testing with the car in motion.

The range of the coverage (600 metres) is enough to cover a circuit of Formula Student competition and we achieve a cheap system that will be used next years to improve the car and its efficiency.

Further work

For the future we need to test the system with the car in motion to check that it satisfy all the team's needs. We have to upgrade the reliability of the system, to avoid failures during the dynamics tests.

And finally, we would like to develop our own graphic interface software, as many teams from the same competition have.